

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Étude de certains comportements de la mémoire humaine

Rinclin, Fabienne

Award date:
1986

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES
NOTRE-DAME DE LA PAIX
NAMUR
INSTITUT D'INFORMATIQUE

ETUDE DE CERTAINS COMPORTEMENTS
DE LA MEMOIRE HUMAINE

Fabienne RINCLIN

Directeur académique :
Mr C. Cherton

Responsables de stage :
Mrs Donnay et Romainville

Mémoire présenté en vue
de l'optention du titre
de licencié-maitre en
informatique.

Année Académique 1985-1986

A

Monsieur Cherton, Promoteur de ce mémoire, qui a bien voulu accepter de diriger cette étude, en remerciement pour sa patience et ses précieux conseils,

Messieurs Donnay et Romainville qui nous ont offert leur aide et leur expérience, à Monsieur Cherton et moi-même,

Toutes les personnes, qui de près ou de loin, m'ont soutenue,

j'adresse mes sentiments de profonde reconnaissance.

Table des matières

	page
- Introduction générale	1
- Première Partie : Projet cadre	3
- Premier Chapitre : Introduction	3
- Second Chapitre : Analyse de tests existants	6
- 2.1 Etude du dialogue	6
- 2.2 Etude du protocole	11
- 2.3 Conclusions	12
- Troisième Chapitre : Spécifications du sous-système utile	14
- Première Section : Objectifs des tests étudiés	15
- Seconde Section : Définitions de différentes notions	16
- 2.1 Les textes	16
- 2.2 Les réseaux de concepts	18
- 2.3 La liaison de paragraphes	20
- 2.4 La hiérarchisation de paragraphes	22
- Troisième Section : Dynamique du logiciel	23
- 3.1 Phase d'introduction d'un test	23
- 3.1.1 Introduction d'explications	23
- 3.1.2 Introduction du matériel de test	26
- 3.1.3 Introduction d'une réponse type	28
- 3.2 Phase d'exécution d'un test	29
- 3.2.1 Présentation d'explication	29
- 3.2.2 Présentation des matériaux	30
- 3.2.3 Réalisation d'une réponse	33
- 3.3 Phase de présentation de la réponse type	35
- 3.4 Phase de recueil d'informations sur une réponse	36
- 3.4.1 Recueil d'informations	36
- 3.4.2 Destruction de réponses ou de tests	39

- Deuxième partie : Découpe hiérarchique et modulaire	40
- Première Section : Spécification des modules	41
- 1.1 Introduction	41
- 1.2 Spécifications	42
Niveau 0	42
Niveau 1	43
Niveau 2	46
Niveau 3	54
Niveau 4	60
Niveau 5	66
Niveau 6	73
Niveau 7	75
7.1 Analyse	75
7.2 Implémentation	80
- Conclusion Générale	102

Annexes

Annexe 1 : Présentation d'un exemple des trois tests spécifiés
Annexe 2 : Présentation graphique de la découpe en modules
Annexe 3 : Spécification des écrans
Annexe 4 : Spécifications des primitives d'accès aux collections de données
Annexe 5 : Présentation de la partie implémentée
Annexe 6 : Manuel d'utilisation

Introduction générale

L'idée qui donna naissance à ce mémoire était de fournir un outil informatique aux pédagogues dans le cadre de leurs études sur le comportement de la mémoire humaine.

Ce projet, qui peut sembler original pour un mémoire en informatique, a pris forme à partir de l'intérêt d'informaticiens pour des sujets concernant l'étude de la mémoire et aussi à partir de l'intérêt de pédagogues vis-à-vis de l'informatique comme moyen d'expérimentation.

Cependant, le comportement de la mémoire humaine est un sujet vaste, ardu et encore peu connu. D'autre part, le temps imparti pour la réalisation du mémoire et le type d'ordinateur choisi (Macintosh), nous ont obligés pédagogues comme informaticiens, à réduire nos ambitions de départ.

L'objectif de départ était de créer un logiciel permettant d'introduire et de réaliser des tests psychologiques sur la mémoire humaine. Les résultats de ces tests auraient été ensuite analysés dans le but de fournir un certain profil de chacune des personnes ayant réalisé un de ces tests. Ce profil permettant, par la suite, de proposer à la personne analysée un entraînement des mécanismes de la mémoire adapté à ses besoins personnels.

Les pédagogues ayant une connaissance encore limitée des mécanismes de la mémorisation, le logiciel proposé dans le projet cadre ne contient pas de partie entraînement des mécanismes et ne comprend qu'une analyse très limitée. Celle-ci se résume à une présentation, à l'écran ou sur papier, des réponses enregistrées. Ces dernières pouvant être accompagnées d'éléments donnant aux pédagogues des informations sur leur déroulement.

L'analyse de l'objectif de départ par rapport aux limitations rencontrées a donc fourni comme résultat le projet suivant :

Réaliser un logiciel permettant :

- d'introduire, dans un ordinateur, des tests psychologiques portant sur la mémoire humaine.
- de soumettre ces tests introduits à différentes personnes et enfin,
- de consulter les résultats des tests réalisés.

L'objectif du logiciel étant déterminé, les pédagogues ont fourni une analyse de tests psychologiques existants qui portent sur les mécanismes de mémorisation. L'insertion de cette analyse dans le cadre de ce mémoire avait deux buts qui étaient :

- de déterminer si :
 - * soit il était intéressant et concevable de créer un logiciel permettant d'introduire et de réaliser les tests existants analysés par les pédagogues.
 - * soit il était préférable de concevoir un logiciel propre à chaque test ou chaque type de tests.
- de présenter aux lecteurs le matériel de base qui a été fourni par les pédagogues pour réaliser ce mémoire.

Les désirs des pédagogues ont ensuite été analysés d'un point de vue informatique dans le but de fournir des spécifications complètes et précises. Ces spécifications ont été réalisées pour trois types de tests de mémorisation qui ont été choisis par les pédagogues. (le lecteur trouvera dans l'annexe 1 une présentation des trois types de tests spécifiés) Une découpe hiérarchique et modulaire a été exécutée à partir de ces spécifications.

Enfin, une partie du sous-système formé par les trois types de tests a été implémentée.

Première partie : Projet cadre

Chapitre 1 : Introduction

L'étude des mécanismes de la mémoire humaine étant un vaste sujet, il a été nécessaire de définir un projet cadre précis dont un sous-système utile sera implémenté dans le cadre de ce mémoire.

Avant de définir l'objet de ce projet cadre, il semble intéressant, pour le mémorant, de présenter l'évolution de l'idée de base du mémoire. Car c'est cette évolution qui a permis d'aboutir au choix d'un objectif précis pour le projet cadre. C'est pourquoi une première proposition d'objectif va être présentée brièvement. Ensuite, les critiques qui ont entraîné l'abandon de cet objectif seront présentées. Enfin, l'objectif adopté pour le projet cadre sera défini.

Le premier objectif était basé sur l'hypothèse qu'il est possible de déterminer le degré d'utilisation effectif et potentiel d'un mécanisme de mémorisation par une personne donnée.

Le logiciel correspondant au premier objectif était constitué :

- d'un modèle général contenant plusieurs mécanismes de mémorisation.
- pour chaque personne testée :
 - * d'un profil "actuel" de la personne testée. Ce profil permet de déterminer comment sont utilisés, par la personne testée, chacun des mécanismes étudiés.
 - * d'un profil "potentiel" de la personne testée. C'est-à-dire, le profil qu'elle pourrait acquérir après avoir suivi un entraînement spécifique à ses besoins. C'est-à-dire : détecter les mécanismes sous-employés. Ces mécanismes sont ceux pour lesquels la différence entre le degré d'utilisation effectif et potentiel est élevée.

Les profils potentiel et actuel sont des espaces à N dimensions. N étant le nombre de mécanismes de mémorisation que contient le modèle général.

L'objectif de ce premier logiciel était :

- d'établir les profils actuel et potentiel d'une personne testée en la soumettant à une série de tests de mémorisation contenus dans le logiciel et en analysant ses résultats.
- de présenter des exercices spécifiques à la personne testée dans le but d'améliorer son utilisation des mécanismes sous employés.
- de tester l'efficacité des exercices spécifiques par une seconde série de tests et en établissant le nouveau profil actuel de la personne testée.

Les critiques émises par les pédagogues sur cet objectif étaient les suivantes :

- Il existe plusieurs mécanismes de mémorisation. Chacun de ceux-ci est complexe en lui-même. Cependant, limiter le projet à l'étude d'un seul de ces mécanismes ne correspondait pas à l'objectif de départ qui était de déterminer le profil "actuel" et "potentiel" de la personne testée par rapport à certains des mécanismes de mémorisation. Malheureusement, étudier plusieurs mécanismes en même temps demandait une grande connaissance et une forte maîtrise des interactions qu'ils peuvent avoir entre eux. Actuellement, les interactions possibles ne sont pas suffisamment connues pour que ce projet ait quelques chances d'être valable.
- Etablir une correspondance entre un test ou un exercice donné et un mécanisme précis de mémorisation n'est pas chose aisée. En général, un test ou un exercice ne peut pas être considéré comme révélant un seul mécanisme de mémorisation mais plutôt comme révélant une fonction de plusieurs de ces mécanismes. De plus, connaître le rôle joué par chacun des mécanismes dans une fonction n'est pratiquement pas réalisable pour l'instant.
- Une autre difficulté est apparue lors de la proposition d'essayer d'améliorer, par des exercices, l'utilisation des mécanismes se révélant les moins exploités par rapport aux possibilités potentielles de la personne testée. Cependant, une bonne exécution d'un exercice se rapportant bien à un ou plusieurs mécanismes donnés n'implique pas que la personne testée sera capable de généraliser le contenu de l'exercice proposé et ainsi qu'elle sera capable de le transposer à une autre situation de même type.

Un second objectif a émergé de ces critiques. Ce dernier a été choisi pour être l'objectif du projet cadre qui sera défini dans ce mémoire. Dans cet objectif, l'accent est mis sur la réalisation d'un logiciel permettant :

- aux pédagogues, d'introduire des tests dans l'ordinateur, portant sur la mémorisation.
- aux personnes testées de réaliser ces tests introduits par les pédagogues.
- aux pédagogues de consulter les réponses données aux tests par les personnes testées.

Le logiciel correspondant à cet objectif est composé de deux interfaces qui sont :

- l'interface pédagogue-ordinateur :
Cet interface permet aux pédagogues :
 - * d'introduire des tests psychologiques portant sur la mémoire humaine,
 - * de consulter les réponses fournies à ces tests ainsi que des informations sur leur déroulement.
- l'interface ordinateur-personne testée :
Cet interface soumettra des personnes à des tests préalablement introduits par les pédagogues.

Chapitre 2 : Analyse de tests existants

Cette analyse a été rédigée sur base document "Grille d'analyse de didacticiels" réalisé par les psycho-pédagogues du centre O.S.E de Namur. Ce document a servi de base de travail pour les pédagogues lors de l'expression de leurs exigences vis-à-vis d'un logiciel ayant l'objectif défini dans le projet cadre.

Les deux objectifs de ce chapitre sont :

- de déterminer si la réalisation d'un logiciel général ayant l'objectif du projet cadre et portant sur l'ensemble des tests de mémorisation relevé dans le document était envisageable ou non.
- de permettre aux pédagogues d'explicitier de façon informelle leurs exigences en ce qui concerne les trois types de tests qu'ils ont choisis pour être implémentés comme sous-système utile du projet cadre.

Ces deux objectifs vont être abordés en étudiant de manière générale

- le protocole, c'est-à-dire: le déroulement chronologique, des tests de mémorisation pour en dégager les parties communes,
- les types de données fournies et reçues dans un tel logiciel.

2.1. ETUDES DES DIALOGUES

2.1.1 Le dialogue entre l'ordinateur et une personne testée

* Envoi d'informations à l'utilisateur-testé :

Le logiciel devra être capable de :

- . présenter un texte :

Le logiciel pourrait donner la possibilité à l'utilisateur d'indiquer certains traitements spéciaux à réaliser sur des parties du texte introduit.

Exemples : indiquer que certains mots doivent apparaître en gras, soulignés, caractères italiques, utilisation de figures géométriques...

. présenter des figures :

Le logiciel devra permettre des manipulations sur ces figures telles que: choix, déplacements, orientations, ...

. présenter des réseaux de concepts :

* Définition informelle d'un réseau de concepts :

un réseau de concepts est formé de un ou plusieurs concepts. Un concept est une suite de caractères. Les concepts peuvent être reliés entre-eux par des liaisons de différents types et porte un nom les identifiants.

(note : la définition précise de réseaux de concepts sera fournie dans le chapitre trois "spécifications du sous-système." Il suffit pour l'instant que le lecteur sache qu'un réseau de concepts est un graphe orienté dont les sommets sont des concepts et les relations des liaisons de différents types. Un exemple de test de type réseau est fourni à l'annexe 1)

* Le logiciel devra permettre de différencier :

- les concepts des liaisons
- les types de liaisons entre-elles.
- les liaisons de même type ayant une signification différente.

* Envoi d'une sollicitation à l'utilisateur-testé :

Ceci se pratique la plupart du temps sous forme d'une ou de plusieurs questions posées à l'utilisateur-testé.

Le logiciel devrait être capable de :

- . poser des questions à choix multiples,

- . poser des questions où la réponse peut appartenir à un ensemble d'éléments :

ex : réponse $1/3 = 2/6 = 18/36 = \dots$

synonymes

- . demander de réaliser des dessins ou figures,
- . Demander de construire un réseau de concepts à partir d'un matériel donné,
- . Demander de hiérarchiser les paragraphes d'un texte donné,
(note : la définition de la hiérarchisation d'un texte sera fournie dans le chapitre trois " spécifications du sous-système")
- . Demander la création de liaisons entre les paragraphes d'un texte donné.
(note : la définition de la liaison des paragraphes d'un texte est donnée dans le chapitre trois "spécifications du sous-système.)

* Réception d'un message de l'utilisateur-testé :

Le système devra être capable de :

- . recevoir les réponses aux questions à choix multiples,
- . recevoir les réponses aux questions à plusieurs éléments possibles comme réponse,
- . percevoir des manipulations de figures,
- . recevoir la demande de sélection de parties de texte, et noter l'ordre chronologique de ces demandes,
- . recevoir un dessin fait à "main levée" par l'utilisateur-testé,
- . recevoir des réseaux de concepts.
- . recevoir la hiérarchisation des paragraphes d'un texte,
- . recevoir les liaisons créées entre les paragraphes d'un texte donné.

* Analyse de la réponse de la personne testée :

- Vérification des réponses aux questions à choix multiples et des questions acceptant un ensemble de réponses :
- Enregistrement chronologique des parties de texte sélectionnées,
- Enregistrements des réponses fournies,
- Vérification de l'équivalence entre l'élément mémorisé et l'élément qui a été fourni par le pédagogue dans le cas des dessins et des réseaux de concepts :
A partir de cela, une analyse de la réponse fournie pourra être effectuée suivant les indications préalablement introduites par le pédagogue.

2.1.2 Le dialogue entre le pédagogue et l'ordinateur

* Envoi d'informations au pédagogue :

Le logiciel devra être capable de :

- . présenter un texte :
- . présenter des figures :
Le logiciel devra permettre des manipulations sur ces figures telles que: choix, déplacements, orientations, ainsi que création.
- . présenter des réseaux de concepts :

* Le logiciel devra permettre de différencier :

- les concepts des liaisons
- les types de liaisons entre-elles.
- les liaisons de même type ayant une signification différente.

* Envoi d'une sollicitation au pédagogue :

Le logiciel devrait être capable de :

- . poser des questions à choix multiples,
- . poser des questions où la réponse est libre,
- . demander d'introduire des dessins à main levée ou des figures,
- . demander de construire un réseau de concepts à partir d'un matériel donné,
- . demander de hiérarchiser les paragraphes d'un texte donné,
- . demander la création de liaisons entre les paragraphes d'un texte donné.

* Réception d'un message du pédagogue :

Le logiciel devra être capable de :

- . recevoir les réponses aux questions à choix multiples,
- . recevoir les réponses libres,
- . percevoir des manipulations de figures,
- . recevoir des figures créées,
- . recevoir la demande de sélection de parties de texte, et noter l'ordre chronologique de ces demandes,
- . recevoir un dessin fait à "main levée" par l'utilisateur pédagogue,
- . recevoir des réseaux de concepts,
- . recevoir la hiérarchisation des paragraphes d'un texte,
- . recevoir les liaisons créées entre les paragraphes d'un texte donné.

2.2. ETUDE DU PROTOCOLE

L'analyse de quelques tests existants a permis de dégager un certain déroulement chronologique commun à ces tests. Ce déroulement est du type :

- Présentation du test :
 - * explication du but de l'exposé,
 - * des résultats possibles,
 - * du déroulement chronologique du test,
 - * du (des) type(s) de questions posées,
 - * de la manière dont il faut répondre à ces questions,
 - * d'un exemple.
- Fourniture du (des) matériau(x) de test :
(s'il(s) existe(nt))
ex : texte ...

- Réalisation du test :
Présentation d'une ou de plusieurs séries de questions avec réception des réponses. L'enchaînement des questions et des réponses peut se dérouler de différentes façons :
 - * l'ordre des questions est défini à l'avance. Cet ordre est appliqué lors de la réalisation du test quelle que soit la nature des réponses données.
 - * l'ordre des questions n'est pas déterminé à l'avance. Il dépend de la réponse fournie à chaque question.

Certains tests étudiés imposent une limite de temps pour donner la réponse à une question. D'autres n'imposent pas une telle limite. Dans ce cas, c'est l'utilisateur qui, en répondant à la question courante, détermine le moment où la question suivante lui est présentée. Certains tests offrent la possibilité de corriger la réponse donnée à une question précédente. D'autres ne permettent pas à l'utilisateur de revenir sur les réponses déjà introduites.

- Evaluation :

- * auto-évaluation grâce à des données fournies telles que : tableaux, réponses-types ...
- * évaluation par un tiers .

2.3. CONCLUSIONS

Ce chapitre a, dans le cadre de ce mémoire, permis de :

- dégager :

- * les différentes parties formant les tests de mémorisation analysés,
- * la dynamique de la réalisation de ces tests.

Ce sont ces différentes parties et cette dynamique qui serviront de référence pour la réalisation du projet cadre et plus particulièrement pour le sous-système modélisé.

- déterminer de manière générale les différentes données utilisées dans des tests psychologiques portant sur la mémoire humaine.

La lecture de cette étude fait remarquer qu'un logiciel portant sur la réalisation de tels tests devra présenter et recevoir des données telles que : textes, figures, q.c.m, réponses ouvertes, réponses à des qcm, hiérarchie, réseaux de concepts.

Sans entrer dans une définition précise de ces données, il est certain que des outils permettant le graphisme, des traitements sur textes seront nécessaires. Le choix d'un type d'ordinateur permettant l'emploi de tels outils est donc primordial.

Comme l'analyse des réponses fournies au logiciel se limitera à l'enregistrement des réponses pour pouvoir les représenter quand un pédagogue désire les consulter, les difficultés liées à la vérification sémantique et/ ou syntaxique n'entrent pas dans le cadre de la réalisation du projet informatique.

Ceci permet aux informaticiens de penser qu'il est envisageable de réaliser un logiciel global ayant pour objectif l'introduction de tests psychologiques portant sur la mémoire humaine et la réalisation de ces tests par des personnes.

Un tel logiciel permettrait de rassembler les différents tests existants sur la mémoire. Les pédagogues pourraient ainsi :

- * soumettre de manière plus aisée des personnes à différents tests sur la mémoire dans un même environnement.
 - * accéder facilement à l'ensemble des réponses de chaque personne ayant réalisé un ou plusieurs tests.
 - * avoir un outil qui ferait passer les tests à leur place. Ce qui leur apporterait un gain de temps appréciable.
- dégager les données qui seront utilisées dans les trois types de tests choisis pour être modélisés dans le sous-système réalisé dans ce mémoire.

Les trois types de tests, qui seront précisés dans la spécification informatique du problème, sont :

- * la réalisation d'un réseau de concepts à partir d'un texte donné,
- * la hiérarchisation des paragraphes d'un texte donné,
- * la création de liaisons entre les paragraphes d'un texte donné.

Le lecteur pourra consulter un exemple de ces trois types de tests dans l'annexe 1.

Chapitre 3

Spécifications du sous-système

Dans ce chapitre, nous allons essayer d'analyser et de définir en détail les différents éléments et les notions qui forment les types de tests choisis ainsi que l'enchaînement de ces différents éléments.

Le chapitre est découpé de la façon suivante :

- une première section présentera l'objectif de chacun des trois tests qui vont être spécifiés.

Une consultation en parallèle de la première annexe est vivement conseillée au lecteur. Ceci rendra sa lecture des sections deux et trois beaucoup plus aisée. Car cette annexe présente un exemple concret des trois types de tests étudiés.

- une seconde section fournira un ensemble de définitions.
- une troisième section présentera les différentes étapes de la marche-à-suivre qui permet :
 - * aux pédagogues d'introduire un test psychologique portant sur la mémoire,
 - * à différentes personnes, de réaliser les tests introduits préalablement,
 - * aux pédagogues, de recueillir des informations sur les tests exécutés.

Section 1 : Objectif

1.1 CREATION D'UN RESEAU DE CONCEPTS à PARTIR D'UN TEXTE

La création d'un réseau de concepts est réalisée par un pédagogue ou une personne testée comme réponse à un test de type 'réseau de concepts'.

L'objectif de ce test est de représenter, à l'aide de matériaux qui sont des concepts et des liaisons, la sémantique d'un texte. C'est-à-dire, représenter les idées de texte et les liaisons qui existent entre elles.

1.2 LIAISON DES PARAGRAPHES D'UN TEXTE

La liaison des paragraphes d'un texte est réalisée par un pédagogue ou une personne testée comme réponse à un test de type 'liaison de paragraphes d'un texte'.

L'objectif de ce test est de représenter, à l'aide de liaisons, les relations qui existent entre les différents paragraphes d'un texte.

1.3 HIERARCHISATION DES PARAGRAPHES D'UN TEXTE

La hiérarchisation de paragraphes d'un texte est créée par une personne testée ou pédagogue comme réponse à un test du type 'hiérarchie de paragraphes d'un texte'.

L'objectif de ce test est d'ordonner les identifiants des paragraphes constituant le texte de travail. L'ordonnancement donné aux identifiants de paragraphes doit mettre en évidence les idées principales du texte. Et pour chaque une de ces idées, mettre en évidence les idées secondaires qui s'y rapportent.

Remarque :

Les réponses aux trois types de test sont donc réalisées à partir d'un texte qui est appelé texte de travail.

De plus, dans le cas de tests de type 'réseau' ou 'liaison', des matériaux seront disponibles pour construire la réponse.

Section 2 : Definitions

2.1 LES TEXTES

- - - - -

2.1.1 Définition

.....

Un texte est formé :

- d'un en-tête (optionnel) lui même formé :
 - * d'un titre (optionnel) : suite non vide de caractères.
 - * d'un contenu (obligatoire) : suite non vide de caractères.

- d'une suite non vide de paragraphes (obligatoire).

Chaque paragraphe est formé :

- * d'un titre (optionnel) : suite non vide de caractères.
- * d'une zone identifiante (obligatoire) : suite de caractères.
- * d'un contenu (obligatoire) : suite non vide de caractères.

Il existe dans le logiciel trois types distincts de textes. Ces types se différencient les uns des autres par l'utilisation qui en est faite et par les éléments qui les forment.

Les trois types de textes sont :

- Les textes explicatifs :

Ces textes sont formés d'une suite non vide de paragraphes formé chacun :

- * d'un titre (optionnel)
- * d'une zone identifiante vide
- * d'un contenu.

- Les textes de travail :

Ces textes sont formés :

- * d'un en tête (optionnel)
- * d'une suite non vide de paragraphes.

- Les textes de réponses :

Ces textes sont formés d'un seul paragraphe formé lui même :

- * d'une zone identifiante vide
- * d'un contenu.

2.1.2 Présentation

.....

Pour les textes comprenant un en-tête, cet élément est toujours présenté avant la suite des paragraphes. L'en-tête est séparée de cette suite par une ligne blanche.

Pour les textes comprenant plusieurs paragraphes, chacun de ceux-ci est séparé du suivant, s'il existe, par une ligne blanche.

Dans l'en-tête ou un paragraphe, le titre, s'il existe, occupe à lui seul une ligne de l'écran. Le contenu de l'en-tête ou du paragraphe est affiché à partir de la ligne suivante.

La zone identifiant un paragraphe est placée à la ligne qui précède le titre du paragraphe, s'il existe. Si non, cette zone apparaît à la ligne qui précède le contenu du paragraphe. Le contenu de la zone identifiante apparaît en caractères inversés. Si l'identifiant n'existe pas, cette zone contiendra une suite de caractères blancs inversés.

3.1.3 Utilisation

.....

- Les textes explicatifs sont introduits par le pédagogue pour être présentés aux personnes testées lors de l'explication d'un test.
- Le texte de travail de chaque test est introduit par le pédagogue. Ce test est présenté à l'écran :
 - * avant la réalisation d'une réponse (avec l'en-tête, s'il existe).
 - * pendant la réalisation de la réponse (sans l'en-tête qu'elle existe ou non.)

Pour les trois types de tests, avant la réalisation d'une réponse, le texte de travail n'est accessible qu'en consultation.

Pendant la réalisation d'une réponse et pour les tests de type réseau, le texte de travail n'est accessible qu'en consultation. Par contre, dans les tests de type liaison et hiérarchie de paragraphes, les utilisateurs peuvent

introduire des éléments supplémentaires dans ce texte pendant la réalisation de leur réponse. Ces éléments sont propres à chaque personnes testées et sont le contenu de la zone identifiant chaque paragraphe.

- Les textes de réponses :

Ces textes sont introduits lors de la réalisation d'une réponse par l'utilisateur pédagogue ou testé.

Lorsqu'il le désire cet utilisateur peut introduire ce type de texte pour expliquer l'introduction d'un élément dans la réponse qui est en cours de création.

Ces textes permettent aux pédagogues et aux personnes testées de recevoir des informations supplémentaires lorsqu'une réponse réalisée par une autre personne qu'eux leur est présentée.

2.2 LES RESEAUX DE CONCEPTS

- - - - -

2.2.1 Définition

.....

Un réseau de concepts est un graphe orienté dont :

- les sommets sont de deux types qui sont appelés "concepts" et "liaison rectangle"
- les arcs sont de deux types qui sont appelés "liaisons flèches" et " liaison sans nom".

De plus,

- à chaque sommet de type "concept" sont associées deux suites non vide de caractères qui identifient chacune le sommet,
- à chaque sommet de type "liaison rectangle" est associé un nombre entier qui l'identifie,
- à chaque arc de type "liaison flèche", sont associées deux suites non vides de caractères qui identifient chacune la liaison.
- un sommet de type "liaison rectagle" est relié à zéro, un ou plusieurs sommets de type concepts par des arcs du type "liaison sans nom",
- un arc de type 'liaison sans nom' ne peut relier qu'un sommet de type 'liaison rectangle' à un sommet de type 'concept'.
- un arc du type 'liaison flèche' ne peut pas relier un sommet de type 'liaison rectangle' à un sommet de type 'concept'.

2.2.2 Sémantique associée à chacun des éléments formant un réseau

- Les concepts :

Chaque concept a une signification qui lui est associée et qui l'identifie parmi les autres concepts et les liaisons de type rectangle et de type flèche d'un même test. Cette signification est décomposée en deux éléments qui l'identifient chacun et qui sont :

- * une première suite non vide de caractères identifiant le concept. Cet élément contient l'expression en langue naturelle de la signification du concept.
- * Le second élément est également une suite non vide de caractères identifiant le concept. Cette suite est plus petite que la précédente et contient un rappel mnémotechnique de la signification du concept.

- La liaison rectangle :

La liaison rectangle a une signification unique qui lui est réservée. Cette signification est : 'regroupement de concepts'. Ceci implique que, pour tous les tests de type réseaux de concepts, aucune liaison de type flèche ni aucun concept ne peut avoir la même signification.

Le problème qui se pose est de pouvoir différencier deux utilisations de la liaison rectangle dans un même réseau. C'est pourquoi un élément est associé à chaque utilisation de cette liaison. Cet élément est un nombre qui détermine dans quel ordre les différentes utilisations de la liaison rectangle ont été insérées dans une réponse de type réseau de concepts.

- Les liaisons de type flèche :

Chaque famille de liaisons de type flèche a une signification qui lui est associée et qui l'identifie parmi les autres familles de liaisons de type flèches et qui l'identifie également par rapport à la liaison rectangle et les concepts d'un même test. Cette signification est décomposée en deux éléments qui l'identifient chacun et qui sont :

- * Une première suite non vide de caractères identifiant la famille de liaisons.

Cette suite contient, en langage naturel, l'expression de la signification associée au groupe de liaisons.

- * une seconde suite non vide de caractères plus petite et qui identifie également la famille de liaisons. Cette seconde suite contient un rappel mnémotechnique de la signification associée à une famille de liaisons.

2.2.3 Présentation

Un réseau de concepts pourra donc être présenté à l'écran comme étant formé des éléments suivants :

- une ou plusieurs suites mnémotechniques identifiants des concepts,
- la représentation graphique de 0, une ou plusieurs liaisons de type flèche, de une ou plusieurs familles, associée chacune à son identifiant mnémotechnique,
- la représentation graphique de 0, une ou plusieurs liaisons rectangle chacune associée à son numéro.

2.3 LA LIAISON DES PARAGRAPHERS D'UN TEXTE

2.3.1 Définition

Réaliser la liaison des paragraphes d'un texte, c'est réaliser un graphe orienté dont :

- les sommets sont appelés "sommets de paragraphe",
- les arcs sont appelés "liaison flèche".

De plus,

- à chaque sommet est associée une suite non vide de caractères identifiante,
- à chaque arc sont associées deux suites non vide de caractères identifiant chacune l'arc.

2.3.2 Sémantique associée à chacun des éléments formant une liaison de paragraphes

- Les "Sommets de paragraphe" :

Chaque "sommet de paragraphe" a une signification qui lui est associée et qui l'identifie parmi les autres sommets de paragraphes et les arcs appelés "liaisons de type flèches". La suite de caractères associée à un "sommet paragraphe" correspond à la zone identifiante d'un paragraphe du texte de travail. Il y a donc une relation biunivoque entre un "sommet de paragraphe" et un paragraphe du texte de travail.

- Les liaisons de type flèche :

Chaque famille de liaisons de type flèche a une signification qui lui est associée et qui l'identifie parmi les autres familles de liaisons et les identifiants de paragraphes d'un même test. Cette signification est décomposée en deux éléments qui l'identifient chacun et qui sont :

- * Une première suite non vide de caractères identifiant la famille de liaisons.

Cette suite contient, en langage naturel, l'expression de la signification associée à la de liaisons.

- * une seconde suite non vide de caractères plus petite et qui identifie également la famille de liaisons. Cette seconde suite contient un rappel mnémotechnique de la signification attribuée à la famille de liaisons.

2.3.3 Présentation

Une liaison de paragraphes pourra être présentée à l'écran ou sur papier comme étant formée des éléments suivants :

- un ou plusieurs identifiants de paragraphes
- la représentation graphique de zéro, une ou plusieurs liaisons flèches associées chacune à la suite courte identifiant une famille de liaison flèche.

2.4 LA HIERARCHIE DE PARAGRAPHERS D'UN TEXTE

- - - - -

2.4.1 Définition

.....

Une hiérarchie de paragraphes est un arbre.

A chaque sommet sont associées deux suites non vides de caractères identifiant chacune le sommet. Ces deux suites sont appelées "sommet de paragraphes" et "hiérarchie".

2.4.2 Sémantique

.....

- Les 'sommets de paragraphes' :

La suite de caractères appelée "sommet de paragraphe" contient la zone identifiante d'un paragraphe du texte de travail. Il y a donc une relation biunivoque entre un sommet de paragraphe et un paragraphe du texte de travail.

- La suite de caractères donnant la hiérarchie

Cette suite de caractères identifie le sommet au quel elle se rapporte et contient la hiérarchie qui a été attribuée à ce sommet.

2.4.3 Présentation

.....

L'arbre formant la hiérarchie est parcouru en utilisant la méthode "depth first search". A chaque sommet rencontré, excepté la racine, les deux suites de caractères le formant sont affichées. La première contient un identifiant de paragraphe. La seconde contient une suite de caractères déterminant la hiérarchie.

Les deux suites formant un sommet sont affichées sur une même ligne.

2.5 LES REPONSES

Il existe pour chaque type de test, trois types de réponses :

- la réponse de la personne testée
- les deux réponses pédagogiques appelées réponse-type et réponse-exemple.

Une réponse, quelle que soit son type est réalisée à partir :

- des matériaux qui ont été choisis par le pédagogue pour être présentés et utilisés lors de l'exécution du test ;
- du texte de travail préalablement introduit.

Section 3 : Dynamique du Logiciel

3.1 PHASE D'INTRODUCTION D'UN TEST

Si le pédagogue veut soumettre une ou plusieurs personnes à un test par l'intermédiaire de ce logiciel, il doit d'abord s'assurer que ce test particulier est bien présent dans l'outil. C-à-d : que ce test a déjà été introduit.

Si le test n'a pas encore été introduit, le pédagogue devra réaliser cette introduction. Ceci implique :

- Une demande d'introduction d'un test n'existant pas dans l'outil en indiquant de quel type est le test à introduire ainsi que son identifiant;
- L'introduction des différents éléments essentiels ou optionnels qui forment ce test.

Les différents éléments dont peut être formé un test sont définis et analysés ci-dessous.

3.1.1 Introduction d'éléments explicatifs

La phase d'introduction d'éléments explicatifs n'est pas obligatoire pour introduire un test dans l'outil. Le pédagogue aura donc la possibilité de décider si oui ou non il faut introduire ces éléments.

Si non : Ces éléments sont communiqués par une autre voie que l'outil à réaliser. Ceci n'est pas à considérer dans cette analyse.

Si oui : Examinons les différents éléments explicatifs qui peuvent être introduits.

Les explications introduites pourront se présenter sous deux formes :

- un texte explicatif seul;
- un texte explicatif suivi d'un exemple (réponse-exemple).

A) Le texte explicatif :

.....

Le texte explicatif répond à la définition de texte donnée dans la première section "définitions". Les éléments qui forment ce test sont déterminés par son type qui est 'explicatifs'.

Manipulations possibles lors de l'introduction d'un texte :

- indiquer l'introduction de chacun des éléments qui peuvent former un texte et dont l'existence dépend du type du texte. Ces éléments sont :
 - * l'en-tête
 - * une suite de paragraphes
- demande pour souligner une suite de caractères ;
- indiquer son accord sur le texte introduit ;
- indiquer la demande de révision pour correction ;
- indiquer que l'introduction est terminée.

REMARQUE :

Les différentes manipulations citées ci-dessus sont accessibles à toute personne (testée ou pédagogue) désirant introduire un texte dans le logiciel et ceci quel que soit le type de ce texte (texte explicatif ou de travail ou réponse).

B) L'exemple :

.....

La forme de l'exemple variera suivant le type du test auquel il se rapporte. Ces trois formes sont :

- un réseau de concepts ;
- une hiérarchisation des identifiants de paragraphes du texte de travail;
- une liaison entre les identifiants de paragraphes du texte de travail.

L'introduction de cet exemple par le pédagogue se fera dans les mêmes conditions et avec les mêmes facilités que lorsque celui-ci introduit sa réponse type ou lors de l'introduction par la personne testée de sa réponse. Car cet exemple est un type de réponse (réponse-exemple).

Ceci implique que l'introduction des différentes parties du test devra être ordonnée (étant donné les éléments nécessaires pour réaliser un test)

En plus des éléments qui forment l'exemple, quel que soit son type, le pédagogue pourra introduire un ou plusieurs textes de type réponse qui expliqueront de parties de l'exemple en spécifiant à quel moment chacun doit apparaître. Ceci permettra aux personnes testées de bien comprendre l'objectif du test et les possibilités auxquelles elles auront accès.

Manipulations permettant la consultation du texte de travail lors de l'introduction d'une réponse :

- * accès au texte;
- * quitter le texte;
- * accès à la page suivante, précédente;
- * Dans le cas de tests de type 'hiérarchie' ou 'liaison' :
 - introduction d'un identifiant pour un paragraphe du texte de travail;
 - choix d'un paragraphe du texte par son identifiant;
 - modification d'un identifiant d'un paragraphe du texte qui n'a pas encore été introduit dans la réponse.

Manipulations possibles lors de la réalisation d'une réponse :

- 1) dans le cas d'un test de type réseaux :
 - * vision des concepts disponibles ;
 - * vision des liaisons disponibles ;
 - * choix d'un concept, d'une liaison ;

- * placement d'un concept, d'une liaison dans la réponse;
- * confirmation de la réponse ;
- * déplacement, destruction d'un concept, d'une liaison dans la réponse;
- * manipulations offertes pour introduire un texte de type réponse;

2) dans le cas d'un test de type liaison entre paragraphes :

- * vision des liaisons disponibles ;
- * choix d'une liaison ;
- * placement d'une liaison dans la réponse ;
- * déplacement, destruction d'une liaison dans la réponse ;
- * manipulations offertes pour introduire un texte de type réponse;
- * placement de l'identifiant de paragraphe choisi dans la réponse;

3) dans le cas d'un test de type hiérarchie de paragraphes :

- * introduction et placement de l'élément hiérarchique choisi
- * déplacement, changement d'un élément hiérarchique placé dans la réponse;
- * manipulations permettant l'introduction d'un texte de type réponse.
- * placement de l'identifiant de paragraphe choisi dans la réponse;

REMARQUE :

Les différentes manipulations citées ci-dessus sont accessibles à toute personne (pédagogue ou testé) voulant introduire une réponse quel que soit son type (réponse exemple, réponse type ou réponse du testé).

3.1.2 Introduction des éléments qui formeront le matériel de test

A) Le texte de travail

.....

Ce texte répond à la définition donnée précédemment.

Ce texte sera introduit en utilisant les manipulations offertes pour permettre l'introduction d'un texte.

B) Introduction d'éléments propres à un type de test

.....

- Dans le cas de tests de type RESEAUX de concepts :

Le pédagogue doit procéder à l'introduction de la signification de chacun des concepts et des types de liaisons qu'il a choisi pour un test donné. Ce sont ces concepts et liaisons qui aideront le pédagogue à introduire son exemple et sa réponse type et qui aideront la personne testée à introduire sa réponse.

A) Les liaisons :

L'introduction de la signification des liaisons pouvant être employées s'effectuera à partir de la présentation des types de liaisons offerts par l'outil et qui sont la liaison rectangle et la liaison flèche. (voir définition des réseaux de concepts)

- Pour les liaisons de type flèche :

La signification choisie par le pédagogue est introduite en deux parties qui sont :

- * une suite non vide de caractères longue et identifiante
- * une suite non vide de caractères courte et identifiante

- Pour la liaison de type rectangle :

La liaison rectangle a une seule signification qui est fixée. Le pédagogue doit donc simplement choisir de permettre ou non l'utilisation de cette liaison.

Manipulations offertes au pédagogue pour choisir le(s) type(s) de liaisons à utiliser pour réaliser une réponse :

- * visualiser les deux types de liaisons disponibles ;
- * choisir un type de liaisons parmi ceux disponibles ;
- * associer à la liaison choisie, si elle est de type flèche, une signification identifiante formée de deux parties identifiantes de caractères.
- * donner son accord sur la signification introduite ou demander sa modification.

Lors de l'introduction de chaque partie de la signification de chaque liaison flèche, une vérification est effectuée pour s'assurer que les deux parties introduites sont bien identifiantes par rapport aux liaisons flèches du test déjà pourvues d'une signification.

B) les concepts :

La signification de chaque concept que le pédagogue introduit pour un test donné sera également formée de deux parties :

- * une suite non vide de caractères longue et identifiante
- * une suite non vide de caractères courte et identifiante

Lors de l'introduction de chaque partie de la signification de chaque concept, une vérification est effectuée pour s'assurer que les deux parties introduites sont bien identifiantes par rapport aux concepts du test déjà pourvus d'une signification.

- Dans le cas d'un test de type HIERARCHIE de paragraphes :

Il n'est pas nécessaire d'introduire des matériaux. La personne réalisant sa réponse introduit elle-même la hiérarchisation voulue avec comme seule contrainte de respecter la syntaxe qui sera définie.

- Dans le cas d'un test de type LIAISONS entre paragraphes :

Le pédagogue devra attribuer à chaque liaison choisie du type flèche une signification. Celle-ci sera, comme dans le cas des liaisons flèches des réseaux, composée de deux éléments.

Les manipulations offertes au pédagogue pour réaliser l'introduction de la signification de chaque liaison utilisable pour créer la réponse à un test sont les mêmes que celles offertes dans le cas des tests de type réseaux.

Lors de l'introduction de chaque partie de la signification de chaque liaison flèche, une vérification est effectuée pour s'assurer que les deux parties introduites sont bien identifiantes par rapport aux liaisons flèches du test déjà pourvues d'une signification.

3.1.3 Introduction d'une réponse-type par le pédagogue :

- - - - -

Le but de l'introduction d'une réponse-type pour chaque test est de présenter à la personne exécutant un test une solution possible à celui-ci. Cette solution est réalisée par le pédagogue pour tout nouveau test lors de son introduction et est présentée à la personne testée après que celle-ci ait réalisé sa réponse au test.

Comme aucune distinction n'est établie entre les différents types de réponses en ce qui concerne la manière de les réaliser, la réponse type sera introduite par le pédagogue en utilisant les manipulations disponibles pour créer un texte, pour consulter le texte de travail et les manipulations disponibles sur les matériaux adaptés à chaque type de test. Le pédagogue aura le choix entre deux types de présentation de la réponse-type : une présentation "en un bloc" ou une présentation "progressive". Les deux types de présentation se différencient par le fait que dans la première, le pédagogue n'a pas introduit de texte explicatif lors de la réalisation de la réponse et que dans la seconde, un ou plusieurs textes explicatifs a (ont) été introduit(s).

3.2 PHASE D'EXECUTION DU TEST

Lorsqu'une personne testée va utiliser le logiciel pour réaliser un test, elle doit d'abord introduire deux données qui sont :

- un élément qui va permettre d'identifier cette personne;
- le test, choisi parmi ceux existant dans le logiciel, qui va être réalisé.

Ces deux données introduites, les différentes parties qui constituent le test vont apparaître à la personne testée suivant un ordre donné. L'évolution dans la réalisation du test doit se faire au rythme de la personne testée. Ce qui veut dire que c'est celle-ci qui détermine le moment où elle veut passer à la phase suivante.

3.2.1 Présentation des explications

Si des explications, correspondant au test choisi pour être réalisé, ont été introduites par le pédagogue, elles vont être présentées à la personne réalisant le test. Si aucune explication n'a été introduite, la personne testée passe automatiquement à la phase suivante.

Les explications à présenter consisteront en :

- un texte-explicatif seul ou
- un texte-explicatif suivi d'une réponse-exemple.

A) le texte :

Le texte-explicatif sera disponible pour l'utilisateur testé autant de temps que celui-ci le jugera utile.

La présentation des éléments constituant le texte explicatif sera celle définie dans le point 1.1.3 présentation d'un texte. Les manipulations de consultation d'un texte seront utilisables.

B) l'exemple de réponse :

Le type de la réponse-exemple (réseau, liaison ou hiérarchie) correspond au type du test à réaliser. La réponse-exemple est présentée sous la même forme que celle qui sera utilisée par la personne testée pour réaliser sa réponse-testé ou le pédagogue pour réaliser sa réponse-type. Cette présentation de la réponse-exemple pourra s'effectuer suivant deux modes qui sont:

- présentation "en un bloc" :
la réponse-exemple est présentée sans texte-explicatif l'accompagnant.
- présentation "progressive" :
les éléments constituant la réponse-exemple sont présentés dans l'ordre de leur insertion dans la réponse par le pédagogue. Certains des éléments insérés sont accompagnés d'un texte-explicatif. Lorsqu'un élément qui est présenté est accompagné d'un texte, ce dernier est affiché à l'écran pour être consulté par la personne testée. L'affichage des autres éléments constituant la réponse est interrompu tant que la personne testée ne signale pas qu'elle a consulté le texte et que la présentation de la réponse peut reprendre. Lorsque toute la réponse est affichée, l'utilisateur décide lui-même le moment où il veut passer à la phase suivante.

3.2.2 Présentation à l'écran des matériaux utilisés pour la réalisation d'une réponse

Les matériaux, s'ils existent, utilisés pour la réalisation d'une réponse quel que soit son type (exemple, type ou testé) sont présentés à la personne testée ou pédagogue lors de la réalisation de cette réponse.

Cependant, il est à noter que le texte de travail sera présenté une première fois, pour qu'elle puisse le consulter, avant qu'elle ne procède à la création de sa réponse. Pendant

cette consultation, les manipulations de consultation d'un texte seront disponibles. Lors de la réalisation de la réponse, le texte de travail sera accessible en mode consultation ou création. Le premier mode est utilisé pour les tests de type 'réseau' le second pour les tests de type 'hiérarchie' ou 'liaison'. Suivant le cas, les manipulations de création et/ou de consultation d'un texte seront disponibles.

Si la personne réalisant une réponse, ne sait pas quelles sont les manipulations possibles elle pourra accéder, soit via le programme soit par un moyen extérieur, à un texte lui fournissant toutes les explications nécessaires.

Lorsque la personne testée ou pédagogue a terminé l'élaboration de sa réponse, elle peut le signaler et ainsi avoir la possibilité d'accéder à la phase suivante ou de corriger cette réponse si elle n'est pas satisfaite

A) Le texte de travail

.....

Avant la réalisation de la réponse, le texte de travail est présenté en consultation sous forme :

- d'un en-tête (s'il existe) séparé de la suite des paragraphes par une ligne blanche.
- d'une série de paragraphes séparés les uns des autres par une ligne blanche. Chaque paragraphe est formé d'un titre (optionnel) et d'un contenu.

Pendant la réalisation de la réponse, la personne réalisant le test aura accès au texte de travail pour consultation et/ou création.

- Dans le cas d'un test de type réseau, seul l'accès en consultation est autorisé.

Le texte sera alors présenté sous la forme d'une suite de paragraphes formé chacun :

- * d'un titre, s'il existe
- * d'un contenu.

(l'en-tête n'est donc plus présenté)

- Dans le cas de tests de type liaison ou hiérarchie, l'accès au texte de travail est autorisé en consultation et en création.

Le texte sera alors présenté sous la forme d'une suite de paragraphes formé chacun :

- * d'une zone réservée à l'identifiant du paragraphe.
- * d'un titre, s'il existe
- * d'un contenu.

(l'en-tête n'est donc plus présenté)

Le contenu de la zone identifiante de chaque paragraphe est introduit par la personne qui réalise sa réponse. C'est pourquoi certaines manipulations de création de texte peuvent être utilisées sur le texte de travail.

B) Les matériaux

.....

- Dans le cas d'un test de type RESEAUX de concepts :

Si la personne effectuant le test doit réaliser une réponse du type réseau de concepts, elle devra avoir la possibilité d'accéder, lors de l'élaboration de la réponse, aux concepts et liaisons utilisables pour ce test en ayant la possibilité d'effectuer les manipulations disponibles sur ces matériaux.

- Dans le cas d'un test de type HIERARCHIE de paragraphes :

La personne testée ou pédagogue doit, dans ce type de test, réaliser une réponse qui est la hiérarchisation de tous les paragraphes du texte de travail. Cette hiérarchisation est basée sur la complexité de l'idée développée dans chaque paragraphe du texte de travail. Pour cela, la personne réalisant sa réponse aura la possibilité d'accéder, lors de la création de celle-ci, au texte de travail pourvu de certaines manipulations de création et des manipulations de consultation de texte.

Lorsqu'elle choisit un paragraphe du texte de travail, la personne qui réalise sa réponse introduit une suite non vide de caractères qui doit identifier le paragraphe parmi les autres paragraphes du texte pour les quels un identifiant a déjà été introduit.

Cet identifiant est introduit dans une zone réservée à cet effet. C'est cet identifiant que la personne manipule pour créer sa hiérarchisation - c-à-d : élaborer sa réponse.

Pour qu'une réponse soit considérée comme une hiérarchie, il faut que :

- * tous les paragraphes du texte (excepté l'en-tête) aient été repris dans la réponse créée ;
- * chaque paragraphe du texte n'apparaisse qu'une seule fois dans la réponse ;
- * chaque paragraphe ait été hiérarchisé.

- Dans le cas d'un test de type LIAISON des PARAGRAPHES :

Si la personne testée ou pédagogue doit créer des liaisons entre les différents paragraphes du texte de travail, elle aura la possibilité d'accéder, lors de la création de sa réponse, aux liaisons disponibles pour le test traité. Les manipulations sur les matériaux lors de la réalisation d'un test de ce type seront disponibles.

Comme dans le cas de la hiérarchisation, la personne réalisant le test doit attribuer un identifiant à chaque paragraphe du texte de travail choisi pour être introduit dans la réponse. C'est l'identifiant attribué au paragraphe choisi qui est manipulé et introduit dans la réponse pour représenter le paragraphe voulu.

Pour qu'une réponse de ce type soit considérée comme une création de liaisons entre paragraphes, il faut que :

- * l'identifiant de chaque paragraphe du texte de travail ait été repris dans la réponse créée.
- * l'identifiant de chaque paragraphe du texte ne doit apparaître qu'une fois dans la réponse.
- * la représentation d'une liaison de type flèche apparaissant dans la réponse doit relier deux paragraphes.

3.2.3 Réalisation d'une réponse

- - - - -

- Dans le cas d'un test de type RESEAU de concepts :

La personne réalisant la réponse aura, au début de la création du réseau, un écran vierge et disponible pour recevoir les différents éléments qui formeront la réponse. (c-à-d : le réseau de concepts)

La personne pourra, grâce aux manipulations citées précédemment, accéder :

* au texte de travail :

une demande de consultation du texte de travail le fera apparaître à l'écran.

Lorsqu' elle n'a plus besoin du texte, la personne le consultant le signale par une commande de consultation et le début de réponse réapparaît non modifié.

* aux concepts :

Si la personne veut voir les concepts pouvant être utilisés pour le test concerné, elle utilise une commande de manipulation des matériaux de test qui aura pour effet de lui présenter ces concepts à l'écran. Lorsque la personne n'a plus besoin des concepts, elle le signale par une commande qui a pour effet de faire disparaître les concepts de l'écran. Celui-ci est alors occupé par la solution ébauchée.

Si un concept est choisi pour être introduit dans la réponse, il disparaît de la liste des concepts disponibles. C'est par le choix de la seconde suite de caractères identifiant un concept donné que celui-ci sera sélectionné pour faire partie de la réponse et que celui-ci sera représenté dans la réponse.

* aux liaisons :

Si la personne testée veut voir les liaisons disponibles pour le test visé, elle utilise une commande des matériaux de test qui aura pour effet de lui présenter ces liaisons à l'écran. Elle les manipule en utilisant les commandes citées précédemment.

C'est par le choix de la seconde suite de caractères identifiant une liaison donnée que celle-ci sera sélectionnée pour faire partie de la réponse et que celle-ci sera représentée dans la réponse.

- Dans le cas d'un test de type LIAISON de PARAGRAPHES :

Une personne réalisant sa création de liaisons, aura, au début de sa création, un écran vierge disponible.

La personne aura des manipulations mises à sa disposition pour accéder aux matériaux disponibles qui sont :

* le texte de travail :

une demande de consultation du texte de travail le fera apparaître à l'écran.

Lorsqu' elle n'a plus besoin du texte, la personne le consultant le signale par une commande de consultation et le début de réponse réapparaît non modifié.

* les liaisons disponibles :

si la personne veut visualiser les liaisons disponibles, elle utilise une commande donnée qui aura pour effet de lui présenter à l'écran les liaisons pouvant être choisies. Ces liaisons ont une signification formée :

C'est par le choix de la seconde suite de caractères identifiant une liaison donnée que celle-ci sera sélectionnée pour faire partie de la réponse et que celle-ci sera représentée dans la réponse.

- Dans le cas d'un test de type HIERARCHIE de paragraphes :

La personne réalisant sa hiérarchie aura, au début de sa création, un écran vierge disponible.

Le pédagogue ou la personne testée pourra avoir accès en consultation et en création au texte de travail grâce aux manipulations citées précédemment.

Lorsque les manipulations sur le texte de travail sont terminées, l'utilisateur le signale et la réponse en cours de création réapparaît.

3.3 PHASE DE PRÉSENTATION DE LA REPONSE TYPE DE PÉDAGOGUE

La réponse-type présentée peut être de trois types différents suivant le type du test effectué.

La présentation de la réponse type du pédagogue est effectuée après la réalisation de sa réponse par la personne testée. La réponse-type est présentée sous la forme utilisée par la personne testée pour réaliser sa propre réponse. La présentation d'une proposition de réponse à pour but de permettre à la personne testée d'accéder à une autre réponse et ainsi effectuer une certaine " critique " de la sienne.

Cette présentation pourra s'effectuer suivant deux modes identiques aux modes de présentation d'une réponse-exemple:

- * présentation " en un bloc " sans explication l'accompagnant ;
- * présentation progressive accompagnée d'explications sous forme de textes.

3.4 PHASE DE RECUEIL D'INFORMATIONS SUR UNE SOLUTION DONNÉE à UN TEST EXISTANT DANS LE LOGICIEL

3.4.1 Recueil d'informations

.....

Toute réponse réalisée est conservée. Ceci permet au pédagogue d'aller consulter une de ces réponses quand il le désire. Pour cela, il devra indiquer l'identifiant de la personne ayant effectué le test et l'identifiant de celui-ci. Cette consultation se fera à l'écran ou à partir d'un listing si le pédagogue demande une impression. Si la présentation se fait à l'écran, il faudra tenir compte du fait que cette réponse peut demander plusieurs écrans. Il devra être permis à la personne qui fait cette consultation de progresser d'un écran à l'autre à son rythme propre.

Les manipulations d'écrans de réponse qui seront offertes sont :

- accès à la page suivante, page précédente
- fin de la consultation

Si le test à traiter est de type RESEAU :

La fin de consultation de chaque élément est déterminée par la personne elle-même. Ceci lui permet d'évoluer à son rythme propre.

Quatre éléments pourront être présentés :

* la liste chronologique des concepts et liaisons qui ont été choisis pour réaliser la réponse. Cette liste se présentera, à l'écran comme à l'impression, sous la forme suivante :

une suite d'identifiants de concepts et de liaisons présentés suivant l'ordre de leur introduction dans la réponse avec une indication permettant de déterminer s'il s'agit d'un premier choix ou d'une modification. L'identifiant de chaque liaison sera accompagné des identifiants du concept origine et cible. Chaque occurrence de la liaison rectangle sera représentée par son numéro d'occurrence et sera accompagnée des identifiants de concepts appartenant à cette occurrence.

- * La représentation du réseau de concepts qui a été créé. Cette représentation aura la forme suivante :
 - à l'écran, même présentation que lors de la création du réseau.
 - pour l'impression, la présentation sera la même que lors de la réalisation du réseau, si cela est possible. Si ce n'est pas possible, une représentation pouvant être adoptée est une liste d'identifiants de concepts avec pour chacun la liste des concepts auxquels ils sont reliés avec la nature et la signification de la liaison. Une liste des occurrences de la liaison rectangle sera également imprimée. Cette liste contiendra pour chaque occurrence l'ensemble des concepts lui appartenant.

 - * Le(s) texte(s) réponse(s) accompagnant celle-ci :

Pour l'impression de ceux-ci, s'il en existe plusieurs, ils sont tous regroupés pour former un seul texte. Chaque paragraphe correspondra à un texte-réponse introduit. Chaque paragraphe contiendra un titre qui sera l'identifiant de l'élément au quel le paragraphe se rapporte.

 - * Une comparaison, à l'écran, entre deux réseaux appartenant au même test pourra être demandée. Il faudra pour cela indiquer cette option ainsi que l'identifiant de chacune des deux personnes ayant réalisé les réseaux à comparer. (une des deux personnes pouvant être le pédagogue qui a introduit le test).
- Cette comparaison pourrait se dérouler de la manière suivante, donner successivement :
- les liaisons et concepts communs aux deux schémas;
 - les éléments qui existent dans le premier et pas dans le second et puis inversement;

Il devra être possible de différencier, par la présentation à l'écran, les éléments qui apparaissent en réponse à un nouveau critère des éléments qui étaient déjà présentés. Cette comparaison aura la même représentation que celle adoptée lors de la construction d'un réseau. C-à-d : une représentation graphique.

Si le test traité est de type HIERARCHISATION de paragraphes

Différents résultats pourront être présentés à l'écran comme sur papier :

- * présentation de la hiérarchie telle que la personne choisie l'a réalisée avec consultation possible du texte de travail muni de l'identifiant associé à chaque paragraphe par la personne traitée. (utilisation des manipulations de consultation de texte)
- * présentation du texte de travail seul avec associé à chaque paragraphe l'identifiant choisi.
Présentation hiérarchisée ou non.
- * la liste chronologique des identifiants de paragraphes qui ont été choisis pour réaliser la réponse avec pour chacun d'eux la hiérarchie attribuée.
Cette liste se présentera sous la forme suivante :
une suite d'identifiants de paragraphes présentés suivant l'ordre de leur introduction dans la réponse avec une indication permettant de déterminer s'il s'agit d'un premier choix ou d'une modification. L'identifiant de chaque paragraphe sera accompagné de la hiérarchie qui lui a été attribuée.
- * Le(s) texte(s) réponse(s) accompagnant celle-ci :
Pour l'impression de ceux-ci, s'il en existe plusieurs, ils sont tous regroupés pour former un seul texte. Chaque paragraphe correspondra à un texte-réponse introduit. Chaque paragraphe contiendra un titre qui sera l'identifiant de l'élément au quel le paragraphe se rapporte.

Si le test traité est de type LIAISONS entre paragraphes :

Voici le type de présentation proposée à l'écran comme sur papier :

- * présentation du texte avec pour chaque paragraphe l'identifiant qui lui est associé par la personne traitée.
- * présentation de la réponse telle que réalisée par la personne traitée.

- * la liste chronologique des identifiants de paragraphes et liaisons qui ont été choisis pour réaliser la réponse. Cette liste se présentera sous la forme suivante :
une suite d'identifiants de paragraphes et de liaisons de type flèche présentés suivant l'ordre de leur introduction dans la réponse avec une indication permettant de déterminer s'il s'agit d'un premier choix ou d'une modification. L'identifiant de chaque liaison sera accompagné des identifiants du paragraphe origine et cible.
- * Le(s) texte(s) réponse(s) accompagnant celle-ci :
Pour l'impression de ceux-ci, s'il en existe plusieurs, ils sont tous regroupés pour former un seul texte. Chaque paragraphe correspondra à un texte-réponse introduit. Chaque paragraphe contiendra un titre qui sera l'identifiant de l'élément au quel le paragraphe se rapporte.

3.4.2 Destruction de réponses et de tests

.....

- Des réponses existantes pourront être détruites suivant différents critères :
 - * supprimer les réponses de toutes les personnes existantes à un test donné (sauf la réponse-type du pédagogue)
 - * supprimer une réponse d'un test donné fournie par une personne testée donnée.
 - * supprimer la réponse d'une personne testée pour tous les tests qu'elle a réalisé.
- Un test existant pourra être détruit par un pédagogue. La destruction d'un test aura comme conséquence la destruction de :
 - * toutes les parties formant ce test,
 - * l'identifiant du test dans la liste des tests existants
 - * tous les types de réponses fournies à ce test.

Deuxième partie : Découpe hiérarchique et modulaire

Cette partie présente la manière dont les différents traitements, qui pourront être exécutés, ont été regroupés pour former des modules logiques de différents niveaux.

Cette présentation sera effectuée en prenant comme modèle la démarche proposée par A. Vanlansweerde (Méthodologie de développement de logiciel).

Le choix de la découpe modulaire et hiérarchique qui va être présentée est dû à la volonté de voir le système sous deux aspects ainsi qu'aux caractéristiques offertes par ces méthodes de conception de logiciel.

Les deux visions sont :

- une vision externe :

Cette vision est basée sur des objectifs différents suivant le type d'utilisateurs du système (pédagogue ou personne testée).

- une vision interne :

Cette vision est basée sur la nature des opérations à effectuer et des ressources nécessaires quel que soit le type de l'utilisateur.

Cette vision explique l'architecture de base du système qui est fondée sur le principe suivant : tant qu'il est possible de ne pas différencier le type de test introduit ainsi que la personne qui est concernée, regroupons les actions nécessaires dans un seul module logique.

En ce qui concerne la méthode de hiérarchisation, son application a eu pour résultat la création de différents niveaux ordonnés possédant chacun différents composants. La relation existant entre composants d'un même niveau ou partant d'un niveau supérieur vers un niveau inférieur est une relation "utilise". (définition de cette relation : Soit deux composants A et B, A utilise B ssi le fonctionnement de A dépend d'une version correcte du composant B.)

Cette relation a pour avantages de faciliter la spécification des composants ainsi que de permettre l'utilisation d'un même composant dans différents contextes.

En ce qui concerne la méthode de modularisation, son application avait pour but de cacher le maximum de décisions conceptuelles à l'intérieur de modules pour les rendre invisibles aux autres modules en relation avec les premiers. Ceci a l'avantage de rendre les modifications beaucoup plus aisées en cas de modifications ou d'agrandissement du logiciel.

Le choix de ce type d'architecture a donc donné comme résultat la découpe qui va être présentée.

Section 1 : Spécification des modules logiques

1.1 Introduction

.....

Les modules sont présentés par niveau hiérarchique. Pour chaque module, sa spécification sera caractérisée par

- ses préconditions :

les conditions devant être vérifiées par les arguments du module avant exécution de celui-ci pour que le module puisse s'exécuter correctement.

- sa fonction :

l'énoncé en français de ce que le module réalise ainsi que les éléments obtenus ou l'action effectuée suite à la réalisation correcte du module.

- ses postconditions :

Les conditions que doivent vérifier les résultats du module à la fin d'une exécution correcte de celui-ci.

- les arguments :

- Les autres modules utilisés par le module spécifié.

1.2 spécification des modules

NOTE

Afin d'éviter de toujours écrire les types possibles des tests, textes et réponses, des abréviations ont été utilisées. Ces abréviations sont :

- pour les tests : 'r' = réseau
 'l' = liaison
 'h' = hiérarchie
- pour les textes : 'r' = réponse
 'e' = explicatif
 't' = travail
- pour les réponses : 't' = type
 'e' = exemple
 'p' = d'une personne testée

Niveau 0

- - - - -

Module Debut

.....

Précondition : /

Fonction :

Donner la choix à l'utilisateur entre :

- introduire un nouveau test,
- réaliser un test existant,
- analyser des réponses, supprimer des réponses ou des tests
- sortir du logiciel.

Si l'utilisateur choisi d'introduire un test ou d'analyser des réponses, le logiciel lui demande de fournir un identifiant de pédagogue existant dans le logiciel. L'utilisateur a droit a deux essais après quoi il est éjecté.

Postcondition :

Si le choix demandé est une réalisation ou un arrêt, il est exécuté;

Si le choix demandé est une introduction d'un nouveau test ou l'analyse de réponses et si l'identifiant de pédagogue fourni est correct le choix est exécuté. Si non, l'utilisateur est éjecté du logiciel.

Le module DEBUT utilise les modules suivants :

- I_Test (niveau 1)
- Realisation (niv 1)
- Analyse (niv 1)

Niveau 1

- - - - -

1) Module I-Test

.....

Précondition :

- le nom de pédagogue fourni est un nom de pédagogue existant dans le logiciel.

Fonction :

Ce module permettra aux pédagogues d'introduire les différents éléments qui constituent un test qui n'est pas contenu dans le logiciel.

- Ces éléments sont :
- Les explications (optionnel)
si elles existent , elles sont constituées d'un texte explicatif et/ou d'une réponse-exemple.
 - Les matériaux :
Ceux-ci seront suivant le type de test:
 - * soit des concepts et liaisons ;
 - * soit des liaisons ;
 - Le texte de travail du test ;
 - une réponse type.

Postcondition :

si un test est introduit :

- le type du test est soit 'r' soit 'l' soit 'h'.
- le nom du test n'appartient pas à la liste des tests existants.
- les éléments suivants ont été introduits :
 - un texte explicatif (optionnel)
 - le texte de travail

- les matériaux dans le cas de test de type 'r' ou 'l'
- un exemple de reponse (optionnel)
- une réponse type

Arguments :

- un identifiant de pédagogue

Le module I_Test utilisera les modules :

- * In-Test (niv 2)
- * Ecran (niv 6)

2) Module Réalisation

.....

Précondition : /

Fonction :

Ce module permettra à la personne qui va réaliser un test de choisir ce test parmi la liste des tests existants.

L'utilisateur doit donc "pointer" l'identifiant du test choisi. Deux situations sont alors possibles :

- * Un identifiant de test contenu dans la liste des tests qui existent dans le logiciel est choisi, un message demande à l'utilisateur de donner son identification puis il peut effectuer le test choisi.
- * La personne choisit de retourner au menu précédent.

Postconditions :

- Si le test a été réalisé, la collection des réponses correspondant au type de test donné contient une réponse en plus dont le type est "t" ;
- Si le test donné est le premier test réalisé par la personne donnée, l'identifiant de cette personne est ajouté dans la liste des personnes testées avec l'identifiant du test réalisé.
- Si la personne donnée a déjà réalisé des tests, on ajoute l'identifiant du test donné à la liste des tests effectués par la personne donnée.

Argument : /

Le module Réalisation utilisera les modules suivants :

- List-Test (niv 5) ;
- In-Rep-Test (niv 2).

3) Module Analyse

.....

Précondition :

- le nom de pédagogue fourni est un nom de pédagogue existant dans le logiciel.

Fonction :

Ce module affichera un message qui permettra aux pédagogues de réaliser différentes actions possibles :

- l'impression sur papier ou l'affichage à l'écran de la réponse à un test déterminé réalisée par une personne déterminée ;
L'utilisateur ayant choisi cette option devra choisir parmi une liste :
 - * l'identifiant de la personne ayant réalisé le test
 - * l'identifiant du test.

- la destruction :
 - * de la réponse à un test déterminé et réalisée par une personne testée déterminée,
 - * de toutes les réponses des personnes testées à un test déterminé,
 - * de toutes les réponses d'une personne donnée,

La liste des personnes ayant réalisé le test ne contiendra pas l'identifiant du pédagogue quel que soit le critère de destruction choisi. Ceci permet d'éviter la destruction de la réponse-type ou de la réponse-exemple du pédagogue qui font partie intégrante de l'énoncé du test.

- supprimer un test déjà introduit.
Ceci a pour conséquences l'élimination :
 - * de toutes les parties existantes de ce test;
 - * de l'identifiant du test dans la liste des tests existants;
 - * de toutes les réponses correspondant au test spécifié.
- Introduire un nouveau nom de pédagogue ou supprimer un nom de pédagogue existant.
- visualiser :
 - * la liste des tests réalisés, triés par ordre alphabétique, avec pour chaque test la liste par ordre alphabétique des identifiants de personnes ayant réalisé ce test.

- * la liste, par ordre alphabétique, des personnes ayant réalisé des tests avec, pour chaque personne, la liste par ordre alphabétique, des tests effectués.

Un message demandera à l'utilisateur d'introduire le choix réalisé entre les deux listes citées ci-dessus.

- Retourner au menu général de la procédure.

Argument :

- le nom de pédagogue

Le module Analyse utilisera les modules :

- List-Rep (niv 5)
- A-Test-imp (niv 2)
- A-Test-ecran (niv 2)
- Del-Test (niv 5)
- Del-Rep (niv 5)
- List_Nom_Ped (niv 5)
- Ecran (niv 6)

Niveau 2

- - - - -

1) Module In-Test

.....

Précondition :

- l'identifiant de pédagogue est un identifiant existant dans le logiciel.

Fonction :

Le module permettra au pédagogue d'introduire un nouveau test dans le logiciel (c-à-d : non encore contenu dans celui-ci). Pour introduire un nouveau test, il faut introduire les différents éléments qui vont le constituer en tenant compte du type de ce test.

Avant l'introduction du test, le pédagogue pourra, s'il le désire, avoir accès à la liste des tests déjà introduits pour vérifier que le test n'existe pas. S'il n'existe pas, le pédagogue introduira l'identifiant et le type du test à créer.

Deux cas peuvent donc se produire :

- Si l'utilisateur essaie d'introduire un identifiant déjà attribué à un test existant, un message lui est envoyé signalant le refus de l'introduction avec sa raison et on lui demande de réintroduire l'identifiant du test. La commande est abandonnée si rien n'est introduit et l'utilisateur se retrouve au niveau du menu.
- Si L'identifiant introduit ne correspond à aucun test existant, on demande à l'utilisateur d'introduire le type du test.

Si le type est 'r' ou 'l' ou 'h', l'introduction est acceptée et peut commencer. Sinon, le logiciel demande l'introduction d'un autre type jusqu'au moment où celui-ci est correct.

Les éléments à introduire sont, par ordre chronologique :

- le texte de travail :

le logiciel envoie un message à l'utilisateur lui demandant d'introduire le texte de travail correspondant au test en cours d'introduction.

Lorsque l'introduction est terminée, l'utilisateur le signale par une commande de fin. Deux cas peuvent alors se produire :

Si le pédagogue est satisfait, il le signale et passe à l'élément suivant à introduire.

Sinon, le pédagogue a la possibilité de corriger le texte introduit et lorsqu'il est satisfait, il signale la fin de la correction et passe à l'élément suivant.

- les matériaux ;

s'ils existent pour le type de test introduit, un message est envoyé au pédagogue lui signalant qu'il va devoir introduire ses matériaux.

- un texte explicatif :

Le logiciel envoie un message à l'utilisateur lui demandant s'il veut introduire un texte explicatif.

* Si oui, le pédagogue l'introduit.

Le texte explicatif introduit consistera en un ou plusieurs paragraphes sans titre et sans zone identifiante. Lorsque l'introduction est terminée, l'utilisateur le signale par

une commande de fin. Deux cas peuvent alors se produire :

Si le pédagogue est satisfait, il le signale et passe à l'élément suivant à introduire.

Sinon, le pédagogue a la possibilité de corriger le texte introduit et lorsqu'il est satisfait, il signale la fin de la correction et passe à l'élément suivant.

* Sinon, il n'y a pas de texte explicatif à introduire, le pédagogue le signale. Ceci lui permet de passer à l'élément suivant.

- un exemple de réponse :

Le logiciel demande à l'utilisateur s'il désire introduire un exemple de réponse.

* Si oui, il introduit celle-ci dans les mêmes conditions que pour une autre réponse à ce même test mais d'un type différent. (voir module In-Rep)

Lorsque l'introduction est terminée, il le signale par une commande de fin. Deux cas peuvent alors se produire :

* Si l'utilisateur n'est pas satisfait de la réponse introduite, il le signale et peut ainsi la modifier jusqu'au moment où il signale qu'il est satisfait.

* Si l'utilisateur est satisfait, il le signale et passe à la phase suivante. Il revient ainsi au menu général du module I-Test.

* Sinon, il n'y a pas introduction d'un exemple, l'utilisateur le signale et passe à l'introduction de l'élément suivant.

- Une réponse type :

Le logiciel envoie un message à l'utilisateur lui demandant d'introduire sa réponse-type.

Le pédagogue introduit sa réponse. (voir module In-Rep)

Lorsque l'introduction est terminée, il le signale par une commande de fin. Deux cas peuvent alors se produire :

* Si l'utilisateur n'est pas satisfait de la réponse introduite, il le signale et peut ainsi la modifier jusqu'au moment où il signale qu'il est satisfait.

* Si l'utilisateur est satisfait, il le signale et passe à la phase suivante. Il revient ainsi au menu général du module In-Maj-Mod.

Postconditions :

SI :

- L'identifiant donné pour le test à introduire n'est pas contenu dans la liste des identifiants des tests existants dans le logiciel.
- le type du test à introduire est soit 'r' soit 'l' soit 'h'.

Alors :

- La liste des identifiants de tests existants après l'exécution du module est la liste avant exécution à laquelle on a ajouté un élément constitué de l'identifiant du test introduit et de son type,
- Les différents éléments qui constituent le test introduit sont contenus dans le logiciel.

Sinon, aucun test n'est introduit.

Argument :

- un identifiant de pédagogue

Le module In-Test utilisera les modules suivants :

- In-Mat (niv 4)
- In-Texte (niv 4)
- In-Rep (niv 3)

2) Module In-Rep-Test

.....

Préconditions :

- L'identifiant du test donné est contenu dans la liste des identifiants de tests existants (donc le test existe dans le logiciel).
- Le type de la réponse à créer est soit 't' soit 'e' soit 'p'.
- L'identifiant de la personne testée.

Fonction :

Le module In-Rep-Test permettra à la personne testée qui a choisi d'introduire une réponse à un test existant d'accéder aux différentes parties qui forment ce test pour réaliser une réponse de type 'p'. Ceci suivant une certaine chronologie qui est la suivante :

* Présentation du texte explicatif :

Si le test explicatif existe, il est présenté à l'écran avec, pour l'utilisateur, la possibilité d'employer les

manipulations suivantes : accès à la page suivante, précédente, arrêt de la consultation.

Si non (le texte explicatif n'existe pas), l'utilisateur passe automatiquement à la présentation de l'élément suivant.

* Présentation de l'exemple :

Si un exemple a été introduit, il est présenté à l'écran (voir le module S-Rep).

Si non, (aucun exemple n'a été introduit), l'utilisateur passe automatiquement à la présentation de l'élément suivant du test traité.

* Présentation à l'écran du texte de travail.

Quand l'utilisateur décide de commencer la réalisation de sa réponse, il le signale par une commande qui a pour effet de lui présenter le texte de travail formé d'un en-tête s'il existe et d'une suite non vide de paragraphes ne contenant pas de zone identifiante. Lorsqu'il a pris connaissance du texte de travail, l'utilisateur le signale et passe à la création de sa réponse.

* Présentation, à l'écran, d'un message demandant l'introduction d'une réponse au test déterminé :

Un message signale à l'utilisateur qu'il va pouvoir introduire sa réponse et qu'il a accès au texte de travail et aux matériaux, s'ils existent.

Quand la personne a terminé la création de sa réponse, deux cas peuvent se présenter :

- la personne est satisfaite de sa réponse et passe à la phase suivante.

- la personne n'est pas satisfaite de sa réponse et veut la modifier :

elle le signale et peut ensuite modifier sa réponse. Lorsqu'elle est satisfaite, elle le signale.

Postconditions :

- l'ensemble des réponses à ce test contient un élément supplémentaire qui est la réponse de la personne testée à un test déterminé,
- le fait qu'un test déterminé est réalisé par une personne déterminée de type testé est enregistré.

Arguments :

- l'identifiant de la personne testée;
- l'identifiant du test;

Le module In-Rep-Test utilisera les modules suivants :

- In-Rep (niv 3)
- S-Rep (niv 3)
- S-Texte (niv 3)
- primitives d'accès aux collections.

3) Module A-Test-Ecran

.....

Préconditions :

- l'identifiant du test à traiter appartient à la liste des identifiants de tests existants dans le logiciel.
- l'identifiant de la personne à traiter appartient à la liste des personnes existantes dans le logiciel(soit testée soit pédagogue).
- la personne à traiter a fourni une réponse au test dont l'identifiant est donné.

Fonction :

Ce module a pour but de fournir des renseignements sur une réponse d'une personne donnée, pédagogue ou testée, à un test donné. La nature des renseignements fournis dépend du type de test :

- dans le cas de réseaux :

Le module permettra de choisir entre les éléments suivants :

- * présenter, à l'écran, la liste des concepts et liaisons choisis dans l'ordre chronologique de leur choix ;
- * présenter, à l'écran, la liste des modifications de concepts et de liaisons dans l'ordre chronologique;
- * présenter, à l'écran, le réseau de concepts réalisé sous forme graphique accompagné de un ou plusieurs textes explicatifs de la réponse traitée s'il(s) existe(nt);
- * présenter, à l'écran, une comparaison entre deux réseaux;
- * retourner au menu du module Analyse.

- dans le cas de liaisons de paragraphes :

Plusieurs possibilités sont offertes à l'utilisateur :

- * Recevoir, à l'écran, le texte de travail muni des identifiants de paragraphes choisis par la personne traitée.

(Ce texte peut être consulté pendant la présentation du second élément)

- * Recevoir, à l'écran, la hiérarchie telle que réalisée par la personne traitée.
- * Retourner au menu du module Analyse.

- dans le cas d'une hiérarchie de paragraphes:

Le module permettra de choisir entre les possibilités suivantes :

- * Recevoir, à l'écran, le texte de travail muni des identifiants de paragraphes choisis par la personne traitée.
(Ce texte peut être consulté pendant la présentation du second élément)
- * Recevoir, à l'écran, la hiérarchisation telle que réalisée par la personne donnée.
- * Retourner au menu du module Analyse.

Postcondition :

La commande choisie a bien été réalisée.

Arguments :

- l'identifiant du test,
- l'identifiant de la personne,
- le type de test,
- le type de la personne.

Le module A-Test-Ecran utilise les modules suivant :

- S-Rep (niv 3)
- Comp-Rés (niv 3)
- S-Chr (niv 3)
- Ecran (niv 6)
- primitives d'accès aux collections des réponses et des chronologies de réponses.

4) Module A-Test-Imp

.....

Préconditions :

- l'identifiant du test à traiter appartient à la liste des identifiants de tests existants dans le logiciel.
- l'identifiant de la personne à traiter appartient à la liste des personnes existantes dans le logiciel (pédagogue ou testée).

- la personne à traiter a effectué le test dont est fourni l'identifiant.

Fonction :

- dans le cas de réseaux :

Le module permettra de choisir entre les éléments suivants :

- * impression, sur papier, de la liste des concepts et liaisons choisis dans l'ordre chronologique de leur choix ;
- * impression, sur papier, de la liste des modifications de concepts et de liaisons dans l'ordre chronologique de celles-ci ;
- * impression graphique, sur papier, du réseau de concepts ;
- * impression, sur papier, d'un texte explicatif du réseau s'il existe ; (s'il existe plusieurs textes, ils sont regroupés par ordre chronologique pour n'en former plus qu'un seul). S'il existe plusieurs textes, chacun est formé d'un seul paragraphe sans titre. Ces paragraphes sont regroupés et un titre est attribué à chacun. Ce titre est l'élément auquel se rapporte le paragraphe traité.
- * retourner au menu précédent ;

- dans le cas d'une hiérarchie :

Le module permettra d'effectuer une des possibilités suivantes :

- * imprimer, sur papier, le texte de travail muni des identifiants de paragraphes choisis par la personne traitée.
- * imprimer, sur papier, la hiérarchie, réalisée sur les identifiants de paragraphes choisis par la personne traitée.
- * impression, sur papier, d'un texte explicatif de la réponse s'il existe ; (si plusieurs textes existent, ils sont regroupés par ordre chronologique pour n'en former plus qu'un seul). S'il existe plusieurs textes, chacun est formé d'un seul paragraphe sans titre. Ces paragraphes sont regroupés pour former un seul texte et un titre est attribué à chacun. Ce titre est l'élément de la réponse auquel se rapporte le paragraphe traité.
- * retourner au menu précédent.

- dans le cas de liaisons entre paragraphes :

Le module permettra de choisir entre les actions suivantes :

- * imprimer, sur papier, le texte de travail muni des identifiants de paragraphes choisis par la personne traitée.

- * imprimer, sur papier, la réponse élaborée par la personne déterminée sur les identifiants de paragraphes choisis par la personne déterminée.
- * impression, sur papier, d'un texte explicatif de la réponse s'il existe ; (si plusieurs textes existent, ils sont regroupés par ordre chronologique pour n'en former plus qu'un seul. S'il existe plusieurs textes, chacun est formé d'un seul paragraphe sans titre. Ces paragraphes sont regroupés pour former un seul texte et un titre est attribué à chacun. Ce titre est l'élément de la réponse auquel se rapporte le paragraphe traité.
- * retourner au menu précédent.

Postcondition :

Si un élément a été choisi pour être imprimé, l'impression a été effectuée. Sinon, retour au menu précédent;

Arguments :

- l'identifiant du test ;
- l'identifiant de la personne ayant réalisé le test
- le type du test

Le module A-Test-Imp utilise les modules

- Imp-Texte (niv 4)
- Imp-Rep (niv 3)
- Imp-Chr (niv 3)

Niveau 3

- - - - -

1) Module In-Rep

.....

Préconditions :

- l'identifiant du test correspond à un test existant dans le logiciel si la réponse à traiter est de type "p" si non, cet identifiant ne correspond pas à un test existant dans le logiciel.
- le type du test à traiter est soit "r" soit "l" soit "h" ;
- le type de la réponse à traiter appartient à l'ensemble des types de réponses possibles. (c-à-d : "p" ou "e" ou "t")
- l'identifiant de personne donné correspond soit à un pédagogue existant (cas d'une réponse de type 'e' ou 't' soit à une personne testée existante (cas d'une réponse de type 'p').

Fonction :

Ce module permettra d'introduire une réponse de type donné et appartenant à un test donné de type déterminé.

- Si le test est du type "r", la réponse introduite sera un réseau.
- Si le test est du type "l", la réponse à introduire sera composé d'identifiants de paragraphes du texte de travail reliés par des liaisons déterminées.
- Si le test est du type "h", la réponse à introduire sera composée d'identifiants de paragraphes du texte de travail donné qui auront été hiérarchisés.

Quels que soient le type du test et le type de la réponse, l'introduction de celle-ci se déroule suivant le schéma suivant:

- Affichage d'un message qui indique que l'utilisateur va pouvoir introduire sa réponse.
- Présentation du texte de travail.
- Réalisation de la réponse avec les matériaux appropriés au test traité et accès au texte de travail.
- Lorsque l'utilisateur a terminé l'introduction de sa réponse, il le signale par un message de fin de réalisation.

Deux choses peuvent alors se produire :

- * Si l'utilisateur est satisfait de sa réponse, il envoie un message le signalant et passe à la phase suivante.
- * Si l'utilisateur n'est pas satisfait de sa réponse, il le signale et la possibilité de corriger cette réponse lui est fournie. Lorsqu'il est satisfait, l'utilisateur le signale et passe à la phase suivante.

Postcondition :

- la collection des réponses contient une réponse de plus. Celle-ci est de type donné, a été réalisée par une personne déterminée et appartient à un test donné.

Arguments :

- l'identifiant du test ;
- le type du test;
- le type de la réponse;
- l'identifiant de la personne;
- le type de la personne

Le module In-Rep utilisera les modules suivants :

- S-Mat (niv 4)
- S-Texte (niv 4)

- Ch-1-Parag (niv 5)
- In-Texte (niv 4)
- les primitives d'accès aux collections de réponses ou exemples correspondant au test donné.

2) Module S-Rep

.....

Préconditions :

- l'identifiant de test appartient à la liste des identifiants des tests existants.
- le type de la réponse est soit "t" soit "e" soit "p".
- l'identifiant de la personne donnée appartient à la liste des personnes connues du logiciel. (testées ou pédagogue)
- Si l'identifiant de la personne donnée ne correspond pas au pédagogue, la personne déterminée doit avoir réalisé le test dont l'identifiant est donné.

Fonction :

Le module permettra la présentation graphique à l'écran d'une réponse. Cette réponse sera de type "t", "e" ou "p".

Cette présentation pourra s'effectuer suivant deux modes :

- * présentation du réseau en un coup ;
- * présentation progressive du réseau accompagnée ou non d'explications sous forme de textes.

Les deux possibilités sont présentées à l'utilisateur accompagnées d'une troisième qui consiste à revenir au menu précédent.

Trois cas peuvent alors se produire :

- * L'utilisateur choisit la présentation en un coup, la présentation s'effectue. Lorsque l'utilisateur a terminé sa consultation, il le signale par une commande de fin. Ceci a pour conséquence de le replacer au niveau du menu du module.
- * L'utilisateur a choisi la présentation progressive;
A chaque présentation d'un texte accompagnant la réponse, le déroulement de l'affichage est arrêté tant que l'utilisateur ne demande pas la reprise du défilement par une commande déterminée.
- * Si l'utilisateur veut quitter ce choix, une commande lui sera également offerte.

- Si le test porte sur la création de liaisons ou d'une hiérarchie entre paragraphes du texte de travail, le module permettra :

* la présentation des identifiants de paragraphes, choisis par la personne testée, reliés par des liaisons.

Durant la présentation de cette réponse, l'utilisateur aura la possibilité d'accéder au texte de travail, muni des identifiants de paragraphes choisis par la personne donnée, grâce à une commande déterminée.

Après la consultation, l'utilisateur peut, par une autre commande, réaccéder à la réponse demandée.

Si l'utilisateur veut quitter ce choix, une commande lui sera également offerte.

* retourner au menu précédent.

Postcondition :

- si une réponse a été demandée, elle a été fournie.

Si non, retour au menu précédent.

Arguments :

- l'identifiant de la personne

- l'identifiant du test.

- le type du test.

Le module S-REP utilise les modules suivants :

- S-Texte (niv 4)

- primitives d'accès à la collection contenant les réponses.

3) Imp-Rep

.....

Préconditions :

- l'identifiant de test appartient à la liste des identifiants des tests existants.

- le type de la réponse est soit "t" soit "e" soit "p".

- l'identifiant de la personne donnée appartient à la liste des personnes connues du logiciel. (testées ou pédagogue)

- Si l'identifiant de la personne donnée ne correspond pas au pédagogue, la personne déterminée doit avoir réalisé le test dont l'identifiant est donné.

Fonction :

Le module permettra l'impression graphique d'une réponse. Cette réponse sera de type "t", "e" ou "p".

Le module Imp_Rep utilisera les primitives d'accès aux collections.

Postcondition :

si une réponse a été demandée, elle a été imprimée. Si non, rien n'est effectué.

4) Module Comp-Res

.....

Préconditions :

- les identifiants des deux personnes correspondent à deux personnes existantes (pédagogue ou testée).
- l'identifiant de test correspond à un test existant;
- les deux personnes ont effectué le test dont est donné l'identifiant.
- le type des deux personnes dont l'identifiant est fourni est soit pédagogue soit testé.

Fonction :

Le module a pour but de présenter à l'écran et de façon graphique une comparaison entre deux réseaux. Cette comparaison pourra être réalisée suivant différents critères qui feront évoluer celle-ci. Les critères seront appliqués suivant l'ordre ci-dessous :

- Afficher les concepts choisis qui sont communs aux deux réseaux.

Afficher les liaisons choisies qui sont communes aux deux réseaux et qui relient deux ou plus de deux concepts communs.

- Afficher les concepts qui apparaissent dans le premier réseau et pas dans le second (et inversement).

Afficher les liaisons qui apparaissent dans le premier réseau et pas dans le second (et inversement).

Les liaisons à afficher sont soit totalement propres au premier (second réseau), soit relient des concepts communs aux deux réseaux et des concepts non communs.

Les éléments qui correspondent à un des critères sont présentés de façon à les différencier d'éléments correspondant à un autre critère. (par des jeux de caractères différents, ...)

Un message est envoyé à l'utilisateur avant de commencer l'affichage de la comparaison des deux réseaux choisis. Ce message a pour but de signaler dans quel(s) type(s) de caractères les éléments correspondants aux différentes parties seront affichés. L'utilisateur utilise une commande spéciale pour signaler que l'affichage peut commencer.

Lorsque l'utilisateur a pris connaissance de la comparaison, il le signale par une commande qui lui permet :

- soit de retourner au menu précédent
- soit d'effectuer une autre comparaison.

Postcondition :

Si une comparaison a été demandée, elle a été effectuée.

Arguments :

- les identifiants des deux personnes et leur type;
- l'identifiant du test à traiter.

Le module Comp-Res utilise les primitives d'accès aux collections des réponses de type "r".

5) Module S-Chr

.....

Préconditions

- l'identifiant de test donné appartient à la liste des tests existants,
- le type du test identifié est soit 'r' soit 'l' soit 'h',
- l'identifiant de personne donné appartient à la liste des identifiants de personnes existantes.
- le type de la personne déterminée est soit 'p' soit 't',
- si le type de la personne est 't', la personne identifiée a réalisé le test donné.

Fonction :

Présenter à l'écran la chronologie d'une réponse à un test donné réalisée par une personne donnée.

Les manipulations suivantes seront disponibles : accès à la page suivante de la chronologie, précédente, arrêt de la consultation.

Postcondition :

la liste chronologique qui correspond à la réponse définie par l'identifiant du test, l'identifiant de personne et le type de celle-ci, a été présentée à l'écran.

Le module S-Chr utilise les primitives d'accès aux différentes collections de données.

6) Module Imp-Chr

.....

Préconditions :

- l'identifiant de test donné appartient à la liste des tests existants,
- le type du test identifié est soit 'r' soit 'l' soit 'h',
- l'identifiant de personne donné appartient à la liste des identifiants de personnes existantes.
- le type de la personne déterminée est soit 'p' soit 't',
- si le type de la personne est 't', la personne identifiée a réalisé le test donné.

Fonction :

Imprimer la chronologie d'une réponse à un test donné réalisée par une personne donnée.

Postcondition :

la liste chronologique qui correspond à la réponse définie par l'identifiant de test, l'identifiant de personne et le type de celle-ci, a été imprimée.

Le module Imp-Chr utilise les primitives d'accès aux différentes collections de données.

Niveau 4

- - - - -

1) Module In-Mat

.....

Préconditions :

- le type du test à traiter est soit "r" soit "l";
- le test à traiter n'existe pas dans les tests contenus dans le logiciel.

Fonction :

Le module permettra l'introduction des différents matériaux qui composent un test donné.

Le type des matériaux à introduire dépendra du type du test donné.

- Si le test porte sur la création d'un réseau de concepts :
L'utilisateur reçoit un message lui demandant d'introduire la signification de chacun des concepts puis un second lui demandant d'introduire les deux parties formant la signification de chacune des liaisons qui seront utilisés pour réaliser le test donné. Lorsque l'utilisateur a terminé l'introduction des concepts, une vérification est effectuée pour déterminer s'il a bien introduit un concept au moins et que celui-ci a été accepté. Un concept est accepté si les deux suites de caractères qui le composent l'identifient bien par rapport à tous les concepts déjà introduits et toutes les liaisons déjà introduites. Un concept ne sera pas accepté s'il possède une seule des deux suites de caractères.

Si aucun concept n'a été accepté, on demande à nouveau à l'utilisateur d'introduire des concepts.

Si oui, l'utilisateur passe à la phase suivante. (introduction des liaisons)

Lorsque l'utilisateur a terminé l'introduction des liaisons, une vérification est effectuée pour déterminer s'il a bien introduit une liaison (flèche ou rectangle) au moins et que celle-ci a été acceptée. Pour qu'une liaison flèche soit acceptée, il faut que les deux suites qui la composent soient non vides et que chacune identifie la liaison par rapport aux autres liaisons déjà introduites et aux concepts déjà introduits.

Si aucune liaison n'a été acceptée, on demande à l'utilisateur de réintroduire des liaisons. Sinon, fin du module.

- si le test porte sur la création de liaisons entre les paragraphes du texte de travail, les matériaux seront des liaisons de type flèche disponibles pour l'utilisateur testé. Un message demande à l'utilisateur d'introduire la signification des liaisons de type flèche sous deux formes identifiantes.

Lorsque l'utilisateur a terminé l'introduction de la signification des liaisons, une vérification est effectuée pour déterminer s'il a bien introduit une signification au moins et que celle-ci a été acceptée. Pour qu'une liaison flèche soit acceptée, il faut que les deux suites qui la composent soient non vides et que chacune identifie la liaison par rapport aux autres liaisons déjà introduites et aux concepts déjà introduits.

Si ce n'est pas le cas, on demande à l'utilisateur de réintroduire des significations de liaisons.

Si c'est le cas, l'utilisateur passe à la phase suivante.

Postcondition :

Tous les matériaux obligatoires pour introduire un test de type donné ont été introduits.

Arguments :

- identifiant du test à traiter;
- type du test à traiter.

Le module In-Mat utilisera les modules suivants :

- In-Cpt (niv 5)
- In-Liai (niv 5)
- Ecran (niv 6)

2) Module In-Texte

.....

Préconditions :

- le type du texte à introduire appartient à l'ensemble des types de texte qui peuvent être introduits; (c-à-d : "t" ou "e" ou "r")
- Le numéro de texte ≥ 1 si le type du texte est "r", num-texte = 0 si non.
- si le type du texte est "r", pers-texte contient l'identifiant de la personne introduisant le texte et Typ-Pers le type de cette personne (pédag. ou testé). Si non, le contenu de ces deux zones est indéterminé.

Fonction :

Le module permettra l'introduction d'un texte répondant à la définition donnée dans les spécifications du problème.

L'introduction s'effectuera de différentes façons suivant le type du texte à introduire :

- Si le texte est du type "explicatif" :

Le logiciel demandera à l'utilisateur-pédagogue d'introduire son texte paragraphe par paragraphe avec pour chaque paragraphe un titre, non obligatoire, et un contenu. L'utilisateur doit donc signaler qu'il veut introduire un texte explicatif. Il doit aussi signaler la fin de l'introduction.

- Si le texte est du type "réponse" :
Le logiciel demandera à l'utilisateur (pédagogue ou testé) d'introduire son texte sous la forme d'un seul paragraphe sans titre.
- Si le texte est du type "travail" :
Le logiciel demandera à l'utilisateur-pédagogue d'introduire le texte de la manière suivante :
 - 1) introduction de l'en-tête (s'il existe) composé d'un titre (non obligatoire) et d'un contenu.
 - 2) introduction de la suite non vide de paragraphe avec pour chacun d'eux, un titre (non obligatoire) et un contenu.

Arguments :

- l'identifiant du test ;
- l'identifiant de la personne introduisant le texte si le type de texte est "r". (pers-texte)
- le type de la personne
- le numéro à attribuer au texte à introduire si le type de ce texte est "r".
- le type du texte.

Le module In-Texte utilisera les modules suivants :

- Verif-Don (niv 6)
- Ecran (niv 6)
- les primitives d'accès à la collection dans laquelle le texte introduit devra être conservé.

3) Module S-Texte

.....

Préconditions :

- l'identifiant de test donné correspond à un test existant dans le logiciel.
- le type du test déterminé est soit "r" soit "l" soit "h".
- l'identifiant de personne donné correspond à une personne existant dans le logiciel (pédagogue ou testé).
- le type de la personne est soit pédagogue soit testé.
- le type du texte est soit "t" soit "r" soit "e"
- Si le type du texte est "r", celui-ci peut avoir un numéro attribué qui est > 0 .
Si le type de texte est "t" ou "e", le numéro attribué au texte est 0.

- le logiciel est soit en phase de préparation à la réalisation d'une réponse, soit en phase de réalisation d'une réponse, soit en phase de présentation de l'explication du test ou de la réponse type.

Fonction :

Le module permettra l'affichage à l'écran d'un texte répondant à la définition donnée dans les spécifications du problème.

Ce texte sera de type donné et sera associé à une personne donnée et pourra être associé à un numéro donné (= à 0 si type texte est "e" ou "t", > 0 si type texte est "r").

Les éléments constituant le texte à afficher à l'écran dépendront du type de texte et de son utilisation. (voir la définition de texte donnée dans les spécifications du sous-système)

Postcondition :

Le texte défini a bien été affiché à l'écran.

Arguments :

- l'identifiant du test
- l'identifiant de la personne
- le type de la personne
- le numéro du texte
- le type du texte
- l'identifiant de la phase courante du logiciel.

Le module S-Texte utilisera les primitives d'accès à la collection contenant les textes appartenant aux tests existants.

4) Module S-Mat

.....

Préconditions :

- si le test est en phase de réalisation, l'identifiant du test donné existe dans la liste des tests existant dans le logiciel. Si le test est en phase d'introduction, l'identifiant n'existe pas dans cette liste.
- le type du test donné est soit "r" soit "l".

Fonction :

Le module permettra la présentation à l'écran des différents matériaux disponibles pour introduire les différents types possibles de réponses.

Postcondition :

Les matériaux disponibles ont bien été affichés à l'écran.

Arguments :

- l'identifiant du test
- le type du test

Le module S-Mat utilisera les modules suivants :

- Ch-Cpt (niv 5)
- Ch-L-Parag (niv 5)

5) Module Imp-Texte

.....

Préconditions :

- l'identifiant du test est contenu dans la liste des identifiants de tests existants dans le logiciel.
- Le type du test est soit "r" soit "l" soit "h".
- Si le type du test est "r", le numéro du texte est > 0
Si le type du test est "l" ou "h", le numéro du texte est $= 0$
- l'identifiant de personne donné correspond à une personne existante dans le logiciel.
- le type de la personne est soit pédagogue soit testé.
- le texte déterminé par les identifiants de test, de personne et le type de la personne existe;

Fonction :

Ce module a pour but l'impression sur papier d'un texte d'un certain type et correspondant à un test donné. Le texte est associé à une personne donnée qui est celle qui a introduit le texte.

Postcondition :

Le texte défini a bien été imprimé.

Arguments :

- l'identifiant du test
- l'identifiant de la personne
- le type de la personne
- le type du texte
- le numéro du texte

Le module Imp-Texte utilisera des primitives d'accès aux collections des textes des tests existants.

Niveau 5

- - - - -

1) Module In-Cpt

.....

Préconditions :

- l'identifiant du test donné n'existe pas dans la liste des tests contenus dans le logiciel.
- le type du test donné est "r";

Fonction :

Ce module permettra d'introduire la signification d'un concept dans les matériaux disponibles pour un test donné. Cette introduction se découpe en deux parties :

- . introduction d'une longue suite de caractères identifiante (30 maximum)
- . introduction d'une courte suite identifiante de caractères (6 max)

Pour chaque concept introduit, une vérification est effectuée sur chacune des deux suites :

- . vérification que la suite longue identifie bien le concept introduit parmi les autres concepts et liaisons du test. Vérification que la suite courte identifie bien le concept par rapport aux autres liaisons et concepts introduits. Si cette condition n'est pas respectée, on demande à l'utilisateur de réintroduire une des deux suites ou les deux suivant les cas.
- . vérification que l'utilisateur n'a pas introduit un concept ayant une des deux suites vides.

Postcondition :

- un concept a été introduit ou signal de fin d'introduction.
- si un concept a été introduit, il a été pourvu d'une signification identifiante.

Arguments :

- l'identifiant du test traité
- le type du test traité
- argument signalant l'arrêt.

Le module In-Cpt utilisera les modules suivants :

- Ecran (niv 6)
- Verif_Don (niv 6)
- les primitives d'accès à la collection contenant les concepts appartenant aux tests existants.

2) Module In-Liai

.....

Préconditions :

- Le type du test en cours d'introduction est "r" ou "l".
- l'identifiant du test en cours de traitement n'est pas contenu dans la liste des tests existant dans le logiciel.

Fonction :

Ce module permettra de :

- donner une signification (sous les formes complète et identifiante) à une liaison de type flèche choisie pour être utilisée;

L'utilisateur reçoit un message qui lui indique qu'il peut introduire la première suite de caractères de la signification de la liaison et ensuite un second message qui lui indique qu'il peut introduire la seconde.

Pour la liaison de type flèche introduite, une vérification est effectuée sur chacune des deux suites :

- . vérification que la liaison introduite n'a pas de suite courte identifiante si elle n'a pas de suite longue identifiante.
- . vérification que la suite longue identifie bien la liaison introduit parmi les autre concepts et liaisons du test. Vérification que la suite courte identifie bien la liaison par rapport aux autre liaisons et concepts introduits. Si cette condition n'est pas respectée, on demande à l'utilisateur de réintroduire une des deux suites ou les deux suivant les cas.

Postconditions :

- une liaison de type flèche a été introduite ou signal de fin d'introduction
- si une liaison introduite, elle a été pourvue d'une signification identifiante.

Arguments :

- l'identifiant du test.
- le type du test.
- signal de fin d'introduction.

Le module In_liai utilisera les modules suivants :

- Verif-Don (niv 6)
- Ecran (niv 6)
- les primitives d'accès aux collections des matériaux.

3) Module Ch-L-Parag

.....

Préconditions:

- Le test en cours d'introduction est de type "l".
- l'identifiant donné pour le test à introduire existe dans la liste des identifiants de tests contenus dans le logiciel.

Fonction :

Ce module permettra de :

- visualiser le type de liaisons disponibles dans le cas d'un test de type "l"
- choisir une liaison parmi celles disponibles

Postcondition :

- Si une liaison est choisie, un argument du module contient son identifiant.

Arguments :

- l'identifiant du test traité.
- l'identifiant de la liaison choisie.

4) Module Choix-Cpt

.....

Préconditions :

- Le test en cours de réalisation est contenu dans le logiciel.
- Le type du test en cours de réalisation est "r".

Fonction :

Ce module permettra de :

- visualiser les concepts disponibles pour réaliser le test en cours de traitement ;

- choisir un concept parmi ceux disponibles ;
- éliminer la visualisation des concepts disponibles à l'écran.

Postcondition :

- si un concept a été choisi, l'identifiant du concept choisi est contenu dans un argument et il ne fait plus partie des concepts disponibles.
si aucun concept n'a été choisi, rien n'est modifié.
L'argument a un contenu indéterminé.

Arguments :

- l'identifiant du test traité.
- la liste de concepts utilisables pour réaliser le test traité.

Le module Choix-Cpt utilise les primitives d'accès à la collection des liaisons disponibles pour le test traité.

5) Module Ch-1-Parag

.....

Préconditions :

- Le test en cours de réalisation appartient à l'ensemble des tests existants dans le logiciel ;
- Le type du test traité est "l" ou "h".

Fonction :

Le module permettra :

- d'introduire l'identifiant d'un paragraphe dans la zone identifiante de celui-ci. Si un identifiant a déjà été donné au paragraphe choisi l'introduction est refusée. Si l'identifiant introduit a déjà été attribué, un message annonce le refus de cet identifiant.
- quitter le texte de travail pour revenir à la réponse en cours d'élaboration.
- choisir un paragraphe qui a déjà reçu un identifiant.

Postconditions :

- Si un identifiant de paragraphe a été introduit, il est associé au paragraphe qu'il identifie et au test qui est en cours de traitement.
- Si un paragraphe a été choisi, les deux arguments de la fonction contiennent son identifiant et son numéro.

Arguments :

- l'identifiant choisi

le module Ch_1_Parag utilise les primitives d'accès aux collections des matériaux.

6) Module Del-Rep

.....

Préconditions :

- l'identifiant du test à traiter fait partie des identifiants de tests existants dans le logiciel ;
- le critère de destruction correspond à une personne testée (donc à son identifiant) ou correspond à l'ensemble des personnes testées.
- si le critère correspond à une seule personne testée, celle-ci fait partie des personnes existantes dans le logiciel.

Fonction :

Ce module permettra de détruire :

- tous les résultats qui correspondent à un test déterminé. Ces résultats sont les réponses données par des personnes testées (et pas celle du pédagogue !) et leur chronologie.
- les résultats d'une seule personne testée choisie au test déterminé.

Postconditions :

- Après exécution du module, l'ensemble des résultats ne contient plus les résultats correspondant au test déterminé et au(x) personne(s) déterminée(s).

Si aucun résultat ne correspond au test déterminé, aucun traitement n'est effectué sur l'ensemble des résultats des tests existants.

Arguments :

- le critère de destruction,
- l'identifiant du test,
- l'identifiant de la personne.

Le module Del-Rep utilisera les modules suivants :

- Ecran (niv 6)
- les primitives d'accès aux collections des réponses de type "testé".

7) Module List-Rep

Préconditions :

- Le critère de tri introduit doit être soit "tests" soit "personnes" soit "itest" soit "ipersonne".
- si le critère correspond à une personne, l'identifiant de personne fourni correspond à une personne existante dans le logiciel.
- si le critère correspond à un test, l'identifiant du test fourni correspond à un test existant dans le logiciel.

Fonction :

Ce module permettra de :

- présenter à l'écran la liste des réponses aux tests existants. Cette liste peut être présentée suivant quatre critères qui sont :
 - * présentation des tests dans l'ordre alphabétique et, pour chacun d'eux, présentation de la liste par ordre alphabétique des personnes les ayant effectués.
 - * présentation par ordre alphabétique des personnes ayant effectué un ou plusieurs tests avec pour chaque personne la liste des tests effectués.
 - * présentation, pour une personne donnée, de la liste des tests qu'elle a réalisés.
 - * présentation, pour un seul test donné, de la liste des personnes existantes ayant réalisé ce test.

Note :

L'utilisateur aura à sa disposition deux commandes : arrêt du défilement et reprise du défilement.

Quand l'utilisateur a terminé sa consultation, il le signale par une commande de fin et il est replacé au niveau du menu précédent.

Postcondition :

- les arguments correspondants au critère choisi ont été présentés à l'écran.

Argument :

- le critère de tri.
- l'identifiant

Le module List-Rep utilise le module Ecran (niv 6)

8) Module Del-Test

.....

Précondition :

- L'identifiant du test à détruire est contenu dans la liste des tests existants dans le logiciel au moment de la demande de destruction.

Fonction :

Ce module permettra de :

- détruire tous les éléments se rapportant au test à éliminer. Le type de ces éléments varieront suivant le type du test. Ces éléments sont :
 - * les matériaux;
 - * toutes les réponses - c-à-d : l'exemple, la réponse-type, la(les) réponse(s) du(des) testé(s) ;
 - * Tous les textes-c-à-d : le texte explicatif (s'il existe), le texte de travail, le(s) texte(s) accompagnant les réponses (s'ils existent);
- mettre-à-jour la liste des tests existants ;
- mettre-à-jour la liste des réponses de tests ;

Postconditions :

- Tous les éléments constituant le test à détruire ont été éliminés;
- la liste des tests existants, après exécution de ce module, est la liste des tests existants avant exécution sans le test à détruire ;
- la liste des réponses des personnes testées aux tests après exécution est la liste des réponses aux tests avant exécution ne contenant plus les réponses correspondant au test à détruire;

Arguments :

- l'identifiant du test ;

Pour supprimer un test existant, le pédagogue devra répondre à un message lui demandant de choisir l'identifiant du test visé. Deux cas sont alors possibles :

- * Si l'identifiant de test choisi appartient à la liste des tests qui existent dans le logiciel, l'utilisateur reçoit un message demandant confirmation.

S'il y a confirmation, la suppression est effectuée. Ensuite retour au menu général de Analyse.

Sinon, il y a annulation de la demande et retour au menu de Analyse.

* Si l'identifiant de test choisi n'appartient pas à la liste des test existant, la demande est annulée.

9) Module In-Mot-Ped

.....

Précondition : /

Fonction :

Ce module permet de :

- supprimer un nom de pédagogue existant
- introduire un nom de pédagogue non existant dans le logiciel.
- annuler la commande.

Postcondition :

La commande qui a été demandée a été effectuée;

Le module In-Nom-Ped utilisera les modules suivants :

- Verif-Don (niv 6)
- Ecran (niv 6)

NIVEAU 6

- - - - -

1) Verif-Don

.....

Précondition :

Le numéro d'appel correspond au numéro d'un écran existant.

C'est l'écran qui correspond à ce numéro qui a fourni les données qui doivent être traitées par le module Verif-Don.

Fonction :

Ce module vérifie la correction sémantique des données fournies par l'écran qui correspond au numéro d'appel.

Postcondition :

Si les données traitées sont correctes, un argument est mis à vrai. Si non, il est mis à faux.

Arguments :

- les données fournies,
- l'argument indiquant la corection des données.

Le module Verif-Don utilise les primitives d'accès aux collections de données.

2) Ecran

.....

Précondition :

Le numéro d'appel donné correspond à un numéro d'écran existant.

Fonction :

Ce module a pour but :

- d'afficher l'écran correspondant au numéro d'appel,
- de recevoir les données introduites par l'utilisateur, si elles existent.

Postconditions :

- l'écran demandé a bien été affiché,
- si des données ont été introduites, elles sont placées dans des arguments du module.

Ce module utilise les primitives d'accès aux collections de données.

Niveau 7

- - - - -

7.1. Analyse

=====

Je vais, dans un premier temps, essayer de déterminer quelles sont les données qui doivent être manipulées par le système. Car ces informations sont fondamentales pour le bon fonctionnement du logiciel.

1. Si on veut que l'utilisateur-pédagogue et l'utilisateur-testé aient la possibilité de savoir quels sont les tests existants dans le logiciel, il faut que l'ensemble de ces tests soit stocké quelque part et que les utilisateurs puissent accéder à cet ensemble pour effectuer des actions plus ou moins spécifiques suivant leurs rôles, telles que : création, consultation, destruction.
Pour un utilisateur, il peut être intéressant de pouvoir accéder aux données suivantes sur un test:
 - son intitulé identifiant
 - son type (réseau, liaison ou hiérarchie)
 - quel pédagogue l'a introduit c'est-à-dire : identifier le pédagogue
 - les éléments optionnels qui le composent (2 éléments sont optionnels : la présentation d'un texte explicatif et la présentation d'un exemple)
2. Si l'utilisateur veut savoir quelles sont les personnes qui ont réalisé un ou plusieurs tests ou quels sont les tests qui ont été exécutés et par qui, il faut que quelque part soient conservées les informations qui permettront d'avoir ces renseignements. Ces informations sont :
 - la liste des identifiants des personnes testées
 - la liste des identifiants des pédagogues
 - pour chaque personne testée, la liste des identifiants des tests réalisés par celle-ci.

3. En ce qui concerne les textes manipulés par le programme, le logiciel doit être capable de déterminer de quel type est le texte traité (explicatif, réponse ou de travail), son numéro et qui l'a créé dans le cas d'un texte-réponse. De plus, il faut savoir à quel test appartient le texte. Les données nécessaires pour pouvoir traiter un texte sont donc :

- son type
- son numéro si le texte est de type réponse
- la personne ayant créé le texte
- le test auquel il se rapporte

4. Les utilisateurs, pédagogues ou testés, utilisent des matériaux de tests pour les tests de type 'réseau' et de type 'liaison'. Les matériaux sont soit des liaisons soit des concepts. Les matériaux introduits par le pédagogue pour un test donné doivent être accessibles aux personnes testées s'ils veulent réaliser un test qui utilise ces matériaux. Les données suivantes sont nécessaires lors de l'introduction de matériaux et lors de l'introduction d'une réponse :

- un élément identifiant le test
- le type du test
- pour chaque matériaux :
 - * son type (concept ou liaison de type flèche ou liaison de type rectangle)
 - * sa signification comprenant deux parties :
 - + un identifiant court
 - + un libellé identifiant de longueur plus grande

5. En ce qui concerne les réponses aux tests; ces derniers étant très différents, les trois types de réponses seront traités de façons différentes. Cependant, on peut remarquer que pour traiter une réponse, quelque soit son type, les données suivantes sont nécessaires :

- un intitulé identifiant le test
- l'identifiant de la personne testée ou pédagogue ayant réalisé la réponse.
- le type de la réponse (exemple, testé ou type)

Dans le cas de tests de type 'réseau', les données formant une réponse sont :

- les concepts utilisés par la personne ayant réalisé la réponse
- les liaisons utilisées par la personne ayant réalisé la réponse.

Dans le cas de tests de type 'liaison' de paragraphes, les données formant la réponse sont :

- les liaisons employées
- les identifiants de tous les paragraphes du texte. Ces identifiants sont introduits par la personne réalisant la réponse traitée

Dans le cas de tests de type 'hiérarchie', les données formant une réponse sont :

- des éléments déterminant la hiérarchie.
- les identifiants de tous les paragraphes du texte de travail. Ces identifiants sont introduits par la personne réalisant la réponse traitée.

6. Pour répondre à la demande des pédagogues, la liste chronologique des éléments choisis lors de la réalisation et de la modification d'une réponse sera accessible. Les données nécessaires sont donc :

- l'intitulé identifiant le test
- l'élément identifiant la personne
- les éléments constituant la réponse
- un indicateur signalant si l'élément fait partie de la réponse initiale ou d'une modification de celle-ci.

REMARQUES :

Des données telles que :

- l'intitulé identifiant le test
- l'élément identifiant la personne testée ou le pédagogue
- le type de test
- les concepts et/ou liaisons introduits par le pédagogue et appartenant à un test donné,

sont soit très souvent nécessaires pour accéder à d'autres données, soit très souvent employées comme données principales. Il serait donc intéressant que ces données soient le plus compactes possible dans le système. Ceci dans

le but d'économiser le plus de place possible. La décision a donc été prise de créer des tables de correspondance pour certaines des données citées ci-dessus. Ces tables établissent une relation entre la donnée représentée sous forme complète et sa représentation abrégée qui est identifiante, soit au niveau du système soit au niveau du test. C'est le programme qui gère l'ensemble de ces tables. La représentation condensée choisie est numérique et est appliquée à toutes les données citées ci-dessus. Cette décision implique des modifications dans les points précédents. Ces modifications sont explicitées dans les points 7, 8 et 9 ci-dessous.

7. Pour chaque test de type 'réseau' ou 'liaison', une table de correspondance est créée. Elle contiendra les concepts et/ou liaisons introduits par le pédagogue et leur identifiant numérique interne qui leur est attribué par le logiciel. Pour accéder aux matériaux d'un test, le programme utilisera l'identifiant numérique du test. Les données contenues dans cette table seront :

- les matériaux (concepts et/ ou liaisons) chacun sous forme de leur signification complète et abrégée.
- pour chacun des matériaux son identifiant interne numérique. La valeur de cet identifiant interne est attribuée par le programme.

Les matériaux seront donc identifiés dans un test donné.

8. Le point 1. va être légèrement modifié pour que la collection contenant les intitulés, les types des tests existants et leurs créateurs possède une donnée supplémentaire sur chacun des tests. Cette donnée est l'identifiant numérique interne de chacun des tests.

Remarque :

- l'identifiant interne, attribué par le système au dernier test introduit, devra avoir été conservé pour qu'il soit accessible directement lors de la demande d'introduction d'un nouveau test. Pour cela, il existe deux solutions qui sont :
 - * soit parcourir la liste des test existants et trouver le dernier numéro attribué.
 - * soit conserver la valeur de l'identifiant interne attribué au dernier test introduit.

- arbitrairement, l'identifiant interne du premier test introduit aura la valeur 0.

9. Le point 2. va être légèrement modifié pour que la collection contenant les identifiants externes des personnes existantes, introduits par ces personnes, dans le système possède une donnée en plus sur ces personnes. Cette donnée est l'identifiant numérique interne de chaque personne qui lui est attribué par le logiciel.

Remarque :

- la valeur de l'identifiant interne attribué à la dernière personne introduite devra être conservée pour permettre l'introduction d'autres personnes par la suite.

Deux solutions sont ici aussi concevables comme au point précédent.

- les pédagogues sont des personnes existantes qui ont des fonctions et des droits différents des personnes testées. C'est pourquoi deux collections distinctes seront créées :

- une collection contenant des renseignements sur les différents pédagogues existants.
- une seconde collection contenant des renseignements sur les personnes testées existantes.

10. Si on reprend le point 6. et que l'on examine les tests de type 'liaison' ou 'hiérarchie', il apparaît que chaque personne attribue un intitulé identifiant à chaque paragraphe choisi. Il est nécessaire que pour chaque réponse les identifiants introduits soient conservés ainsi que leur correspondance avec un paragraphe du texte de travail. Les données suivantes sont donc nécessaires :

- l'identifiant interne du test
- l'identifiant interne de la personne ayant réalisé la réponse
- pour chaque intitulé identifiant, un élément permettant d'identifier le paragraphe du texte de travail auquel il se rapporte.

7.2. Implémentation

=====

7.2.0 Introduction

Pour les différents fichiers nécessaires au bon fonctionnement du système, des choix d'implémentation ont dû être réalisés. Ces choix sont dus à l'existence de différentes contraintes imposées par le langage utilisé ainsi qu'aux possibilités offertes par le type de machine choisi. Ces contraintes sont :

- Les enregistrements contenus dans un fichier doivent être de taille fixe et d'un seul type.
- Les méthodes d'accès autorisées sur un fichier sont l'accès séquentiel et l'accès direct.

Il est à remarquer que ces contraintes ne limitaient pas l'implémentation à celle qui a été choisie par le concepteur et que celui-ci a tenté de poursuivre certains objectifs qui sont :

- essayer, lorsque cela est possible, de choisir les implémentations les plus simples.

Cette décision a deux conséquences :

- 1) création de nombreux fichiers de moyenne ou petite taille à la place de gros fichiers.

*dans le cas de l'enregistrement des réponses de tout type, chaque réponse est conservée dans un fichier qui lui est propre.

*ceci est également le cas pour les fichiers contenant les matériaux de tests, s'ils existent; chaque test possède son propre fichier qui lui est propre.

*chaque texte introduit dans le système est, quelque soit son type, enregistré dans un fichier qui lui est propre

*pour chaque réponse réalisée, un fichier contient les matériaux dans l'ordre chronologique de leur choix.

Chaque fichier créé sera identifié par un nom qui lui sera attribué par le système (voir le détail de l'implémentation de chaque collection)

2) le système accédera, chaque fois que cela sera possible , de manière directe à l'enregistrement voulu. Ceci implique que si des numéros d'identifiant doivent être attribués aux enregistrements, ils auront une relation biunivoque avec l'emplacement de ces enregistrements. Cette relation sera de la nature suivante : le numéro qui identifie un élément correspondra à sa place dans le fichier le contenant. Ceci est valable pour les fichiers des personnes testées et pédagogues, pour les fichiers des intitulés de tests, pour les fichiers de matériaux. Il est à remarquer que le premier élément d'un fichier reçoit le numéro 0.

Cette décision pourra, comme il le sera expliqué plus largement par la suite , entraîner une certaine perte de place.

-essayer de limiter le moins possible la taille de certains éléments qui a du être fixée. (ce qui implique l'emploi de constantes)

Exemples :

- * la taille des identifiants de personnes, de concepts, de liaisons et de paragraphe a été limitée à 6 caractères.
Constante : Lid = 6
- * la taille des titres de paragraphes, des intitulés complets de liaisons ou de concepts a été limitée à 20 caractères.
Constante : Ltitr = 20
- * le nombre de réponses données par une personne testée à des tests existants dans le logiciel est de 10. Si la personne testée veut réaliser plus de 10 tests, une réponse, au moins, devra être supprimée dans la liste des réponses fournies par cette personne.
Constante : Nbtest = 10

7.2.1. Collection des tests existants

Le choix qui a été fait est de créer un fichier appelé List_Test qui contient une série d'enregistrements appelés Test. Chaque enregistrement est formé :

- d'un intitulé identifiant de test : In_Test
- d'un type de test : T_Test
- d'un numéro de test interne et identifiant : Num_Test
- d'un élément indiquant si le test existe ou non : Exist

- d'un numéro interne identifiant le pedagogue ayant introduit le test : Num_Ped
- d'un élément permettant de savoir si les deux parties facultatives d'un test(explication et exemple) sont présentes ou non.

Le troisième élément Num_Test découle du choix expliqué dans l'analyse concernant l'emploi de l'identifiant de test. Dans tout le programme, l'identifiant utilisé sera le numéro attribué au test par le programme.

Il est très important de remarquer que les tests seront placés dans le fichier suivant leur ordre de création et que la décision a été prise de faire correspondre le numéro identifiant du test avec son numéro de place dans le fichier. Ceci a l'avantage de permettre un accès direct à partir du numéro identifiant du test traité (ce qui permet un gain de temps) et comme désavantage qu'on ne peut supprimer physiquement un test du fichier. Ceci a eu pour conséquence l'ajout de la notion d'existence d'un test qui permet de déterminer si un test du fichier est ou n'est pas accessible en tant que test existant. Ce désavantage implique une possibilité de perte de place lors de la suppression d'un test existant.

Le type du test est déterminé par un entier. Si la valeur de T_Test est :

- 1 : c'est un réseau
- 2 : c'est une liaison
- 3 : c'est une hiérarchie.

List_Test = file of Test

Test = record

```

    In_Test : string[Lid];
    T_Test : integer;
    N_Test : integer;
    Exist : boolean;
    N_Ped : integer;
    Comp : array[1..2] of boolean;
end;
```


7.2.2 Collection des personnes existantes

La décision a été prise de créer un fichier contenant l'ensemble des personnes testées. Ce fichier est appelé F_Pers et contient des enregistrements appelés Pers. Chaque enregistrement est formé :

- d'un libellé identifiant la personne : Id_Pers
- d'un numéro interne identifiant la personne : N_Pers
- d'une liste de tests réalisés : L_Test
- d'un indicateur d'existence de la personne : Exist

Un second fichier sera créé et contiendra les identifiants de l'ensemble des pédagogues. Ce fichier est appelé F_Ped et contient des enregistrements appelés Ped. Chaque enregistrement est formé :

- d'un libellé identifiant le pédagogue : Id_Ped
- d'un numéro interne identifiant le pédagogue : N_Ped
- d'un indicateur d'existence du pédagogue : Exist

Deux fichiers distincts ont été créés pour les personnes testées et les pédagogues car les fonctions et les privilèges de chacun sont différents. Cette distinction permet d'avoir une vue plus claire du système ainsi qu'un accès plus rapide aux données appartenant à ces deux fichiers.

Le second élément, dans les deux fichiers, a été créé, comme dans le cas des tests, pour identifier les personnes à l'intérieur du programme de la façon la plus compacte possible. Le numéro attribué à une personne correspond à sa place dans le fichier. Cette décision a les mêmes avantages et inconvénients que la décision similaire prise pour les tests existants.

Chaque enregistrement définissant une personne testée indique également les tests réalisés par celle-ci. Le nombre de réponses pouvant être enregistrés pour une personne testée est limité à 10. Il sera possible de supprimer l'indication de la réalisation d'une réponse d'une personne testée à un test donné. (dans le cas de suppression de cette réponse)

Pour le fichier des personnes testées, un élément de la liste des tests réalisés, L_Test, est soit non existant, c-à-d ne contient pas d'indication sur un test réalisé, soit existant, c-à-d contient le numéro identifiant interne d'un test réalisé existant.


```

F_Pers = file of Pers
Pers : record
    Id_Pers : string[Lid];
    N_Pers : integer;
    L_Test : array [Nbtest] of Pers_Test;
    Exist : boolean;
end;
Pers_Test : record
    Exist : boolean;
    N_Test : integer;
end;

F_Ped = file of Ped
Ped : record
    Id_Ped : string[Lid];
    N_Ped : integer;
    Exist : boolean;
end;

```

7.2.3 Collection des concepts et liaisons introduits

La décision a été prise que pour chaque test de type 'réseau' ou de type 'liaison', un fichier serait créé pour contenir les matériaux se rapportant au test traité. (matériaux = concepts et/ou liaisons) Il y a autant de fichiers matériaux que de tests de type réseau ou liaison. Ce choix est du au fait qu'il permet d'avoir une implémentation claire, plus simple que si tous les matériaux des différents tests étaient regroupés dans un même fichier et plus rapide au point de vue temps d'accès.

Le nom donné au fichier correspondant à un test sera identifiant et formé comme suit :

F_Mat_Numéro id. du test

exemple : F_Mat_10 est le fichier contenant les matériaux du test portant le numéro 10.

Chaque fichier contiendra :

- la signification du matériau sous deux formes:
 - * l'intitulé long identifiant
 - * l'intitulé identifiant court

- le numéro identifiant le matériau de façon interne dans le test considéré;
- un élément indiquant si l'élément est accessible(c-a-d : existe)

Dans le cas de test de type réseau, le fichier des matériaux contiendra d'abord les concepts introduits puis les liaisons disponibles. (avec la liaison rectangle si elle est employée)
 Pour chaque test de ce type, le nombre de concepts disponibles sera mémorisé dans un fichier appelé F_N_Con.

F_Mat_N = file of Mat

Mat = record

Sign_Mat : string[Ltitr];

Id_Mat : string[Lid];

N_Mat : integer;

Exist : boolean;

end;

F_N_Con = file of N_Con

N_Con = record

N_Test : integer;

Nb_Con : integer;

end;

7.2.4 Collections des types de textes

A chaque texte créé correspondra un fichier. Cette décision a été prise pour les mêmes raisons que la décision de créer un fichier pour les matériaux de chaque test. (voir point précédent)

Le nom du fichier permet d'identifier le texte. Ce nom est composé de deux, trois ou quatre éléments significatifs :

T_N.test_type du texte_N.texte_N.pers

Les deux derniers éléments (numéro de texte et de personne) ne sont présents que dans le cas où le texte traité est de type réponse.

Les différents types de textes déterminent la valeur de l'élément type texte. Si c'est un texte :

- de travail, type texte = 1
- explicatif = 2
- réponse = 3

exemples : T_11_2 désigne le fichier explicatif du test numéro 11.

T_11_3_1_4 désigne le nom du fichier contenant le premier texte de la réponse de la personne ayant le numéro 4, pour le test numéro 11.

Les textes de travail seront formés :

- d'un en-tête (s'il existe) contenant ou non un titre;
- d'une suite non vide de paragraphes avec ou sans titre.

Les textes explicatifs et réponses seront formés d'une suite non vide de paragraphes avec ou sans titre.

Tous les fichiers de textes seront de type caractères. Les caractères spéciaux suivants permettront de délimiter les différents paragraphes et en-têtes :

- pour le titre de l'en-tête : #
- pour l'en-tête : @
- pour un paragraphe : §
- pour un titre de paragraphe : f
- pour souligner une partie de texte : ^

7.2.5 Collections des réponses aux types de tests existants

Trois types de fichiers réponses distincts correspondront aux trois types de tests existants. C'est à dire qu'à chaque type de réponse correspondra un fichier contenant des enregistrements ayant une structure propre.

Pour chaque réponse, quel que soit son type, un fichier est créé. Ce fichier est identifié par son nom qui est composé des éléments suivants :

typ-rep_N. test_N. pers

typ-rep indique de quel type est la réponse:

- si typ-rep contient la valeur : RP, c'est une réponse d'une personne testée.

- si typ-rep contient la valeur : RT, c'est une réponse type.
- si typ-rep contient la valeur : RE, c'est une réponse exemple.

exemple : RP_15_7 est la réponse donnée au test numéro 15 par la personne testée identifiée par le numéro 7.

REMARQUE :

- le lecteur remarquera que l'identifiant d'une réponse ne comporte pas la notion de version c'est-à-dire une même personne ne peut effectuer deux fois au plus le même test si la (les) réponse(s) précédente(s) n'est (ne sont) pas détruite(s). Cette restriction porte également pour les identifiants de fichiers chronologiques ainsi que pour la liste des tests réalisés par une personne testée (fichier des personnes testées).

A) Collections des réponses de type 'liaison de paragraphes'

+++++

Il faut tenir compte dans les réponses de types 'liaison' et 'hiérarchie' que la personne qui réalise la réponse introduit des identifiants de paragraphes qui lui sont propres. Pour chaque réponse différente, il y a donc une liste différente d'identifiants de paragraphes. La notion de table de correspondance ne permet pas de gagner de la place ici. Si cette solution avait quand même été adoptée, le nombre de tables de correspondance aurait été équivalent au nombre de fois que des tests de type 'liaison' ou 'hiérarchie' sont réalisés. C'est pourquoi la décision a été prise d'introduire la chaîne de caractères représentant l'identifiant dans la réponse et non pas un numéro qui l'identifierait.

Les objectifs qui devront être respectés par l'implémentation choisie sont :

- pouvoir faire le lien entre un identifiant de paragraphe choisi et le paragraphe du texte visé.
- ne pas limiter le nombre d'identifiants de paragraphes (et donc de paragraphes pour un texte de travail).
- permettre la destruction d'identifiants de paragraphes et de liaisons.

- permettre l'introduction d'un texte explicatif lors de l'insertion dans la réponse d'un nouvel identifiant ou d'une liaison.

- ne pas limiter le nombre de liaisons possibles entre les identifiants de paragraphes.

De cet objectif a découlé l'idée de créer un chaînage avant entre les enregistrements d'un fichier. Ce chaînage existerait entre deux ou plus de deux enregistrements quand le nombre de liaisons partant d'un identifiant de paragraphe est trop grand pour que toutes les informations nécessaires (type de liaison et identifiant cible) soient contenues dans un seul enregistrement. Cette solution permet donc de ne pas limiter le nombre de liaisons possibles à partir d'un identifiant de paragraphe origine.

Nous allons donc trouver deux types d'enregistrements dans un fichier réponse pour un test 'liaison'. Ces deux types ne se différencieront pas par la forme des enregistrements mais par leur contenu :

- * le premier type d'enregistrements contiendra un identifiant de paragraphe origine (avec les données nécessaires sur cet identifiant).

- * le second type contiendra un nombre d'identifiants cibles (avec des données nécessaires sur ceux-ci) d'un même identifiant origine. (Ce type d'enregistrement appartient donc à un chaînage)

- permettre un accès rapide à une réponse donnée et à un élément donné de celle-ci.

L'implémentation suivante a été adoptée :

1) cas d'un enregistrement contenant un concept origine :

Un tel enregistrement est créé lors de l'ajout d'un identifiant de paragraphe à la réponse traitée.

Les informations suivantes doivent être enregistrées :

- * l'identifiant du paragraphe (introduit par la personne)
- * le numéro du paragraphe (pour faire la liaison avec le paragraphe du texte de travail)
- * un numéro de texte explicatif, s'il existe.

2) cas d'un enregistrement appartenant à un chaînage :

Un tel enregistrement peut être créé ou recherché lors de l'ajout d'une liaison dans la réponse traitée.

Le but recherché est d'avoir la possibilité d'insérer, dans le fichier réponse, les informations nécessaires suivantes : quel est l'identifiant origine de la liaison

introduite, quelle est cette liaison, quel est l'identifiant cible de la liaison, y-a-t-il un texte associé à cette liaison et si oui, lequel.

Pour cela, le système recherche l'enregistrement contenant l'identifiant du paragraphe origine. Deux cas sont alors possibles :

- * l'enregistrement contenant l'identifiant du paragraphe origine a encore des zones disponibles pour contenir les informations à enregistrer sur la nouvelle liaison. Ces zones vont être remplies avec les trois informations indiquées ci-dessus.

- * L'enregistrement contenant l'identifiant du paragraphe origine n'a plus de zone disponible pour contenir les informations sur la nouvelle liaison.

Le système analyse alors le dernier élément de cet enregistrement.

Cet élément est un entier qui est soit égal à zéro soit négatif.

Si le dernier élément est égal à zéro, cela veut dire que l'enregistrement est juste rempli et qu'il faut, pour enregistrer les nouvelles données, créer un chaînage. Pour cela, le système va créer un nouvel enregistrement à la fin du fichier et va placer l'adresse de cet enregistrement dans le dernier élément de l'enregistrement contenant l'identifiant origine. La chaîne est ainsi créée. Il faut encore que les informations sur la nouvelle liaison soient enregistrées dans l'enregistrement créé en fin de fichier.

Si le dernier élément est positif, le système parcourt la chaîne jusqu'au moment où il trouve une place vide ou un dernier élément d'enregistrement nul.

Remarque :

- Si on veut supprimer un identifiant de paragraphe de la réponse, deux cas sont à prévoir :

- * soit l'identifiant à supprimer est origine de liaison(s). Alors, il suffit de trouver l'enregistrement où l'identifiant est origine et de rendre inaccessible l'enregistrement. Si une chaîne existe à partir de ce record, il suffit de la parcourir et chaque fois que le logiciel arrive à un record rendre celui-ci inaccessible.

- * soit l'identifiant à supprimer est cible de liaison(s) : Il faut pour chaque enregistrement existant, examiner les

différents éléments qui pourraient contenir l'identifiant recherché. Si un des identifiants examinés correspond à l'identifiant à supprimer, les zones impliquées (c-à-d : contenant l'identifiant, le type de liaison et le texte éventuel) doivent être rendues vierges.

- Si on veut supprimer une liaison de la réponse, il suffit de partir de l'identifiant origine de la liaison et de rechercher dans la liste des paragraphes cibles celui qui est visé par la suppression. Cet élément ainsi que le numéro de la liaison et un texte explicatif éventuel doivent être éliminés de cette liste (c'est-à-dire : les parties de l'enregistrement qui sont concernées doivent être rendues vierges).

Composition des enregistrements :

Le fichier réponse contient une suite d'enregistrements appelés Rep_1.

Chaque enregistrement est formé des éléments suivants :

/Exist//typ_O /id§-C-O /N1_C /N. §-R/N2 //N_R_§C /N.-R/N3 // Ptr/

- L'enregistrement pourrait être considéré comme formé de quatre parties. La première contenant le premier élément qui est un indicateur d'existence de l'enregistrement, la seconde partie les éléments deux, trois, quatre, cinq et six dont le contenu est variable, la troisième partie les éléments sept, huit, neuf traitant de l'identifiant du paragraphe cible et la quatrième partie le dernier élément qui indique la présence ou l'absence de chainage.
- Le fait que la partie 2 possède 2 types d'informations possibles est du à la volonté de ne pas répéter inutilement l'identifiant de paragraphe d'origine dans cette partie.

Composition :

- le premier élément indique si le record est accessible ou non : Exist.
- le second élément est un indicateur donnant le type du premier identifiant de paragraphe contenu dans le record :

- Typ_O (le type de l'identifiant est soit cible soit origine). Typ_O est vrai si le second élément est un identifiant de paragraphe origine, faux sinon.
- le troisième élément est une zone contenant un identifiant de paragraphe origine ou cible(dans ce cas Id_C_O est = ' ').
 - le quatrième T_N1_C est une zone pouvant contenir un numéro de texte associé à l'élément précédent si celui-ci est un identifiant de type origine. Si aucun texte n'est associé à l'introduction de l'identifiant origine, la zone est égale à 0. Si l'identifiant est de type cible, la zone contient le numéro de paragraphe cible.
 - le cinquième élément est une zone contenant soit un numéro de paragraphe soit un numéro de relation suivant que le type de l'identifiant de paragraphe Typ_O indique que l'identifiant est origine ou cible : N_R_\$.
 - le sixième est une zone pouvant contenir un numéro de texte associé à l'introduction dans la réponse de l'élément précédent si celui-ci est un numéro de relation : T_N2.
Si aucun texte n'est associé à l'introduction de cette relation, la zone est égale à 0.
 - le septième élément N_R_\$\$C est une zone contenant le numéro de paragraphe cible (= 0 s'il n'y en a pas).
 - le huitième élément est une zone contenant le numéro de la relation par laquelle Id_C est visé : N_Rel.
 - le neuvième est une zone contenant un numéro de texte associé au huitième élément (c-à-d : un numéro de relation) : T_N2.
Si aucun texte n'est associé à l'introduction de cette relation, la zone est égale à 0.
 - la dixième et dernière zone contient un pointeur permettant de constituer une chaîne d'identifiants de paragraphes cibles d'un même identifiant de paragraphe origine : Ptr.

Rep_L record

```

    Exist   : boolean;
    Typ_O    : boolean;
    Id_C_O   : string[Lid];
    T_N1_C   : integer;
    N_R_$   : integer;
    T_N2     : integer;
    Id_C     : string[id];
    N_Rel    : integer;
    T_N3     : integer;
    Ptr      : integer;

```

end;

Avantages et inconvénients de cette implémentation :

- Le nombre d'identifiants de paragraphe est illimité.
- Le nombre de liaisons possibles pour un identifiant origine est illimité.
- la destruction d'identifiants de paragraphe ou de liaison est possible. L'implémentation choisie peut occasionner de la perte de place car lors de la destruction d'un identifiant de paragraphe, celui-ci n'est pas physiquement détruit.
- un chaînage est très vite créé. (Dès qu'un identifiant est origine de plus d'une liaison) Ceci permet de limiter la perte de place qui peut apparaître si le dernier enregistrement d'une chaîne n'est pas rempli. (ici la perte se réduit à une liaison supplémentaire possible)
- pas de répétition de l'identifiant de paragraphe origine. Ceci a pour conséquences un gain de place du à la non redondance mais également une implémentation un peu plus difficile à comprendre. Ce désavantage est du au fait que des mêmes zones d'enregistrement peuvent contenir deux types différents d'informations.
- l'introduction de textes explicatifs est permise et l'implémentation permet de retrouver ces textes.

B) Collection des réponses de type 'hiérarchie'

+++++

Même remarque que pour la collection de réponses de type liaison en ce qui concerne les identifiants de paragraphes. Il n'existe donc pas de table de correspondance pour les identifiants de paragraphes.

Les objectifs que l'implémentation choisie devra respecter sont :

- permettre un accès rapide à une réponse donnée et dans le cadre d'une réponse, à des informations données.
- pouvoir faire le lien entre un identifiant de paragraphe et le paragraphe concerné dans le texte de travail.
- ne pas limiter le nombre de paragraphes du texte de travail.
- pouvoir associer un texte explicatif au choix d'un identifiant de paragraphe et de l'attribution à cet identifiant d'une place dans la hiérarchie.

Implémentation choisie :

Le fichier d'une réponse de type hiérarchie est formé d'enregistrements Rep_H dont chacun est composé des éléments suivants :

/ Exist /Identifiant-§ / Num-§ / Hiérarchie / Num-texte /

Rep_H record

```
    Id_§   : string[Lid];  
    N_§    : integer;  
    Hier   : string[Lid];  
    N_Text : integer;  
    Exist  : boolean;  
end;
```

Composition des enregistrements :

- le premier élément est un identifiant de paragraphe :
Id_§
- le second élément est le numéro du paragraphe du texte de travail auquel l'identifiant se rapporte : N_§
- le troisième élément est alphanumérique et détermine la hiérarchie choisie pour ce paragraphe : Hier
- le quatrième élément est un numéro de texte, si un texte explicatif a été introduit pour expliquer la hiérarchisation du paragraphe traité : N_Test
- le cinquième élément est un indicateur d'existence de l'enregistrement.

Avantages et inconvénients de cette implémentation :

- C'est une implémentation très facile à comprendre. Cependant, comme les identifiants de paragraphes sont placés dans l'ordre dans lequel ils ont été choisis, il n'y a aucun tri basé sur la hiérarchie ou sur l'ordre des paragraphes dans le texte de travail. Ceci peut entraîner un temps plus long pour permettre la présentation des identifiants de paragraphes dans le texte de travail ou suivant la hiérarchie qui leur a été attribuée.

C) Collection des réponses de types 'réseaux'

+++++

Objectifs à respecter par le logiciel :

- * Le logiciel doit enregistrer les éléments suivants :
 - le concept origine
 - le concept cible (en spécifiant la liaison et le concept origine)
 - la liaison rectangle utilisée soit :
 - * comme groupement de concepts
 - * comme "concept" cible ou origine de liaison flèche

On peut remarquer que la liaison rectangle possède un rôle ambigu qui peut la faire considérer dans certains cas comme une liaison et dans d'autres cas comme un concept.

- * Permettre un accès rapide aux données formant une réponse.
- * Ne pas limiter le nombre de concepts ou de liaisons.
- * Pouvoir associer un texte explicatif lors de l'introduction d'un concept ou d'une liaison dans la réponse.

Implémentation choisie :

Pour comprendre l'implémentation qui a été choisie, il faut tenir compte des éléments cités ci-dessous. Ceci va nous amener à différencier quatre types d'enregistrements :

- enregistrement contenant un concept origine, ses concepts ou rectangles cibles et les liaisons flèches utilisées.
- enregistrement contenant une occurrence d'une liaison de type rectangle et les concepts lui appartenant.
- enregistrement contenant une occurrence d'une liaison de type rectangle qui est l'origine de liaisons de type flèche. La cible de chaque liaison de type flèche étant soit un concept soit une occurrence de liaison rectangle. Chaque enregistrement contient également un élément permettant d'identifier les liaisons flèches utilisées
- enregistrement appartenant à un chaînage :
 - * soit partant d'un concept origine (donc contient des concepts cibles ou des occurrences de liaisons rectangles cibles avec les liaisons)
 - * soit partant d'une occurrence de liaison rectangle créée (donc contient la liste des concepts appartenant à cette occurrence)

- * soit partant d'une occurrence de liaison rectangle considérée comme "concept" origine (donc contient la liste des concepts et/ou les occurrences de liaisons rectangles cibles ainsi que les liaisons de type flèche utilisées)

Il est à remarquer qu'un enregistrement ayant comme élément de départ une occurrence d'une liaison rectangle peut être de deux types différents selon que l'occurrence vient d'être créée ou qu'une liaison flèche entre cette occurrence et un autre élément vient d'être ajoutée à la réponse. (ceci implique que l'on peut trouver dans des enregistrements différents une même occurrence de liaison rectangle)

Ces différents types d'informations vont se retrouver dans des enregistrements formés des mêmes éléments. Ceci implique qu'une zone pourra contenir différentes informations suivant les six types d'enregistrements cités plus haut.

Composition de l'enregistrement :

.....

La représentation suivante a donc été adoptée. Le fichier réponse contient une suite de records appelés Rep_R.

Chaque enregistrement est formé des éléments suivants :

```
-----
/ Exist // typ / Rec_cc_co / N / N_Txt // Tableau // Ptr /
-----
```

- le premier élément indique si le record est accessible ou non : Exist.
- le second élément (Typ) est un indicateur qui par sa valeur donne le type des deux éléments suivants qui sont Rec_cc_co et N.
 - Si Typ = 1
 - * Ele contient un numéro de concept origine, N = 0.
 - Si Typ = 2
 - * Ele contient un numéro d'occurrence de la liaison rectangle considérée comme "concept origine", N est le numéro de la liaison rectangle.
 - Si Typ = 3
 - * Ele contient un numéro d'ocurrence de la liaison rectangle créée, N contient le numéro de la liaison rectangle.

- Si Typ = 4

- * Ele contient soit un numéro de concept cible si le nombre contenu est ≥ 0 , soit un numéro d'occurrence de liaison rectangle cible si le nombre contenu est < 0 . N contient un numéro de relation de type flèche.

- Si Typ = 5

- * Ele contient un numéro de concept contenu dans une liaison rectangle. N = 0.

Remarque :

Si Typ = 4 ou 5, l'enregistrement fait partie d'un chaînage. Dans le premier cas, c'est un chaînage à partir d'un concept origine ou d'une occurrence de la liaison rectangle considérée comme "concept origine". Dans le second cas, c'est un chaînage à partir d'une occurrence qui vient d'être créée de la liaison rectangle.

- Le cinquième élément contiendra un numéro de texte explicatif (> 0) associé à l'introduction du troisième élément si ce texte existe. Si non, N_Txt = 0.

Dans le cas où Typ = 2 ou 5, N_Txt = 0 obligatoirement car aucun texte explicatif ne peut être associé au troisième élément.

- le sixième élément est une zone contenant un tableau. Chaque élément de ce tableau est composé de trois parties.

- * si le troisième élément est un concept origine (Typ=1), les parties du tableau seront respectivement :

- + un entier qui est un numéro de concept cible (≥ 0) ou d'occurrence de la relation rectangle (< 0).
- + un entier qui est un numéro de relation flèche
- + un entier qui est un numéro de texte (s'il existe)

- * si le troisième élément est une occurrence d'une liaison rectangle créée (Typ = 3), les parties du tableau seront respectivement :

- + un entier qui est un numéro de concept appartenant à une occurrence de liaison rectangle
- + un entier qui est un numéro de concept appartenant à une occurrence de liaison rectangle
- + un entier de valeur nulle

- * si le troisième élément est une occurrence d'une liaison rectangle origine d'une liaison flèche (Typ = 2), les parties du tableau seront respectivement :
 - + un entier qui est soit un numéro de concept cible (≥ 0), soit un numéro d'occurrence d' une liaison rectangle (< 0)
 - + un entier qui est un numéro de relation flèche
 - + un entier qui est numéro de texte (s'il existe)
- * si le troisième élément fait partie d'un chaînage, les parties du tableau seront respectivement :
 - + Dans le cas d'un chaînage à partir d'un concept ou d'une occurrence de la liaison rectangle origine (Typ = 4), les types des différentes parties du tableau sont identiques au cas où le troisième élément est un concept origine ou est une liaison rectangle considérée comme un concept.
 - + Dans le cas d'un chaînage à partir de concepts appartenant à une occurrence de la liaison rectangle (Typ = 5), les types des différentes parties du tableau sont identiques au cas où le troisième élément est une nouvelle occurrence de la liaison rectangle.
- la septième et dernière zone contient un pointeur permettant de constituer une chaîne d'identifiants de paragraphes cibles d'un même identifiant de paragraphe origine : Ptr.

Rep_R record

```

    Exist : boolean;
    Typ   : integer;
    Ele   : string[id];
    N     : integer;
    N_T   : integer;
    Tab   : array[1..3] of integer ;
    - Ptr : integer;
end;
```


Avantages-inconvénients de l'implémentation choisie

- une même occurrence d'une liaison rectangle peut apparaître dans deux enregistrements différents :
 - + à la création : un enregistrement indiquant les concepts appartenant à l'occurrence de la liaison "rectangle".
 - + lors de la création d'une liaison flèche partant de l'occurrence de la liaison rectangle.

Les concepts origines n'apparaissent qu'une seule fois.

- l'implémentation choisie est très compliquée et est donc difficile à comprendre. Cependant elle permet un enregistrement compact avec un grand degré de liberté et également un nombre illimité de concepts et de liaisons flèches ou rectangles.
- le tableau comporte trois éléments qui peuvent contenir soit :
 - + deux numéros de concepts appartenant à une occurrence de la liaison rectangle.
 - + une liaison de type flèche avec le numéro de relation et le numéro du concept ou rectangle cible.

La perte maximum de place, avec ou sans chaînage, se présente dans le cas d'un tableau vide.

7.2.6 Collection contenant le dernier numéro attribué à un test et le dernier attribué à une personne testée ou pédagogue.

Ce fichier a pour nom F_Num. Il contient trois entiers dont le premier est le dernier numéro attribué à un test, le second le dernier numéro attribué à une personne testée existante et le troisième le dernier numéro attribué à un pédagogue existant.

7.2.7 Collections contenant les matériaux de réponse dans l'ordre chronologique de leur choix pour chaque personne ayant réalisé un test.

La décision qui a été prise est de mettre la chronologie de chaque réponse créée dans un fichier qui est propre à cette réponse. Le nom du fichier permettant d'identifier le texte. Ce nom est composé de la manière suivante :

Chr_N·TEST_N·PERS

Le numéro de personne sera précédé de la lettre P si la personne traitée est un pédagogue ou de T si la personne traitée est une personne testée ou de E si la réponse correspondant à la chronologie est un exemple.

Exemple :

le fichier contenant la chronologie de la réponse de la personne testée identifiée par le numéro interne 8 ayant réalisé le test 25 aura comme nom :

Chr_25_P8

Chaque fichier chronologique contient les matériaux de tests choisis lors de la première réalisation du test et lors de sa correction (Ex : déplacement, changement de hiérarchie, élimination, ...)

L'organisation envisagée pour chaque fichier chronologique est séquentielle. C-à-d : tout matériau, lorsqu'il est choisi, est placé en fin de fichier. Les éléments choisis lors de la création de la réponse apparaissent donc dans l'ordre de leur ajout dans la réponse. Ensuite, on peut trouver les matériaux introduits lors de la modification de la réponse. Ceux-ci sont aussi dans l'ordre de leur ajout. Une zone de chaque enregistrement indique si celui-ci fait partie de la première ou de la seconde réalisation de la réponse. Cette zone était au départ un booléen. Cependant, il a semblé au concepteur qu'un entier serait plus commode si les pédagogues veulent un jour permettre plusieurs modifications d'une réponse.

Cette organisation a été choisie car pour restituer la chronologie de l'utilisation des matériaux, il suffit de parcourir séquentiellement le fichier.

La composition des enregistrements pour chaque type de test est différente et est présentée ci-dessous.

Pour les tests de type hiérarchie

- - - - -

Les fichiers chronologiques contiendront des enregistrements formés des éléments suivants :

- une zone contenant un numéro de paragraphe : N_§
- une zone indiquant si c'est un premier choix ou une première, seconde, ... correction : N_Choix
- une zone indiquant la hiérarchie attribuée : Hier

Chr_H record

```
    N_§      : integer;
    N_Choix   : integer;
    Hier      : string[1d];
end;
```

Pour les tests de type liaisons

Les fichiers chronologiques contiendront des enregistrements formés des éléments suivants :

- une zone contenant un numéro de paragraphe soit introduit dans la réponse soit origine d'une liaison flèche :
N_Ele1.
- Une zone contenant un numéro de paragraphe cible si Ele1 est un numéro de paragraphe introduit.
- Une zone contenant un numéro de relation si Ele1 est un numéro de paragraphe origine d'une relation ou une zone qui est nulle si Ele1 est un numéro de paragraphe qui vient d'être introduit dans la réponse.
- une zone indiquant si c'est un premier choix ou une première, seconde, ... correction : N_Choix

Chr_L record

```
    N_Ele1    : integer;
    N_Ele2    : integer;
    N_Rel     : integer;
    N_Choix   : integer;
end;
```

Pour les tests de type réseau

Les fichiers chronologiques contiendront des enregistrements formés des éléments suivants :

- une zone (N_Ele1) qui contiendra soit :
 - * un numéro de concept ajouté dans la réponse
(N_Ele1 >= 0)
 - * un numéro d'occurrence de la liaison rectangle ajoutée dans la réponse (N_Ele1 = - (num.occ) < 0).
 - * un numéro d'un élément origine d'une liaison flèche.
Cet élément origine sera soit :
 - un concept (N_Ele1 > 0).
 - une occurrence de liaison rectangle
(N_ele1 = - (num.occ) < 0)

- une zone (N_Ele2) qui :
 - * dans le cas où le premier élément (N_Ele1) est un numéro de concept ajouté ou un numéro d'occurrence de liaison rectangle ajouté, cette seconde zone est nulle.
 - * dans le cas où le premier élément est un numéro d'élément origine d'une liaison flèche, cette seconde zone est un numéro d'élément cible qui est soit :
 - un concept (N_Ele2 >= 0)
 - une occurrence de liaison rectangle (N_Ele2 = -(num.occ) < 0)
- une troisième zone N_Rel contenant :
 - * dans le cas où les deux premières zones contiennent un élément origine et un élément cible, le numéro de la relation flèche employée.
 - * dans les autres cas, une valeur indéterminée.
- une zone N_Choix indiquant si c'est un premier, second, troisième, ... choix.

REMARQUE :

- Dans le cas d'une liaison de type rectangle, l'information sur les concepts appartenant à cette occurrence de la liaison n'est pas enregistrée. Ceci pour ne pas compliquer la structure de l'enregistrement. Pour avoir cette information, il suffit de consulter la représentation graphique de la réponse. Pour le cas où dans une même réponse, plusieurs occurrences de la liaison rectangle apparaissent, celles-ci seront différenciées par un numéro leur étant attribué.

Ch_R record

```

N_Ele1 : integer;
N_Ele2 : integer;
N_Rel  : integer;
N_Choix : integer;
end;
```


Conclusion générale

Ce mémoire a permis d'établir qu'il était réalisable du point de vue informatique de créer un logiciel général regroupant un ensemble de types de tests psychologiques sur la mémoire humaine. Ce logiciel a comme objectif celui qui a été défini dans le projet cadre. (première partie, chapitre deux) Pour mémoire, cet objectif est de fournir aux pédagogues un logiciel qui leur permettra :

- d'introduire des tests psychologiques portant sur la mémoire humaine,
- faire exécuter ces tests par différentes personnes,
- accéder aux résultats des tests effectués.

Pour l'instant, sans l'utilisation du logiciel général, la présence des pédagogues est nécessaire lors de la réalisation d'un test par une personne. Cette présence est indispensable pour fournir des explications et pour recueillir des informations sur la façon dont est réalisé le test par une personne donnée. Lorsque le logiciel général sera implémenté, ce travail sera réalisé par le logiciel. Ceci permettra aux pédagogues de ne plus être présent continuellement lors de la réalisation de tests. Ils seront donc disponibles pour d'autres tâches telles que l'analyse approfondie des réponses fournies aux tests.

Trois types de tests avaient été choisis par le pédagogue pour être implémentés dans un sous-système utile. Ces tests ont chacun un objectif qui est la création :

- d'un réseau de concepts à partir d'un texte ou
- d'une liaison des paragraphes d'un texte ou
- d'une hiérarchisation des paragraphes d'un texte.

Les parties du sous-système qui ont été implémentées dans le cadre de ce mémoire sont :

- le niveau des primitives d'accès au fichier (niveau 7 de la découpe hiérarchique),
- les écrans et la procédures de vérification sémantique des données,
- Les modules appartenant à la partie 'Introduction d'un test' sauf le module d'introduction d'une réponse (In_Rep).

Les parties 'Réalisation de test' et 'Analyse' doivent encore être implémentées.

Pour la partie 'Analyse', certains problèmes de présentation graphique des résultats devront encore être résolus. Celui qui semble le plus important de ceux-ci, concerne la présentation graphique d'un réseau de concepts enregistré qui a été par le logiciel. (c-à-d : la consultation graphique de réseaux qui ont été réalisés auparavant) Les enregistrements des données formant un réseau au niveau des fichiers ne contient pas de renseignement sur la position où était placé un élément appartenant à la réponse lors de la création de celle-ci. Le problème qui se pose est d'avoir une présentation graphique d'une réponse qui soit "lisible". C'est-à-dire : une réponse dans la quelle on peut facilement distinguer les concepts et les liaisons avec pour chaque liaison de type flèche, son origine et sa cible.

Trois solutions sont envisageables :

- enregistrer tel quel, c-à-d : sous forme graphique, le réseau de concepts. Ceci est tout à fait réalisable grâce à des commandes existantes dans le pascal du macintosh. Cependant, cette solution a comme gros désavantage d'occuper énormément de place sur une disquette.
- enregistrer les données sur un réseau de concepts comme il est prévu de le faire au niveau fichier. C'est-à-dire, en utilisant l'implémentation choisie. Et donner au logiciel la possibilité de présenter le réseau de manière graphique grâce à l'utilisation d'une ou plusieurs méthodes qui permettent d'obtenir un graphe plan.
Cette solution semble assez complexe et demandera sans doute une étude approfondie.
- modifier l'enregistrement des données au niveau des fichiers pour enregistrer la position de chaque élément lors de la création d'un réseau.

Lorsque le sous-système utile sera complètement implémenté, des extensions pourront être réalisées pour obtenir l'entièreté du logiciel général. Ceci veut dire que d'autres types de tests seront insérés dans le sous-système réalisé. Ces autres types de tests sont ceux qui ont été précisés dans l'analyse de tests existants (première partie, deuxième chapitre).

Annexes

Annexe 1 : Présentation de trois types de tests

Cette annexe va présenter les matériaux et les réponses des trois types de tests qui ont été choisis par les pédagogues pour être implémentés dans le sous-système utile.

Ces trois types de tests sont :

- la réalisation d'un réseau de concepts
- la réalisation de la liaison des paragraphes d'un texte
- la réalisation de la hiérarchisation des paragraphes d'un texte.

Ces trois types de tests sont réalisés à partir d'un texte appelé texte de travail. Le texte employé pour présenter les trois exemples est présenté à la page suivante.

LE TEXTE DE TRAVAIL

(Extrait du cours d' "Histoire des Temps Modernes" de M. G Philippart de Foy).

Luther mène une vie religieuse stricte et reste néanmoins angoissé sur son sort dans l'au-delà. Peut-on vraiment mériter le salut par ses oeuvres ? C'est la relation mécanique entre les bonnes oeuvres et le salut qui paraît suspecte à Luther.

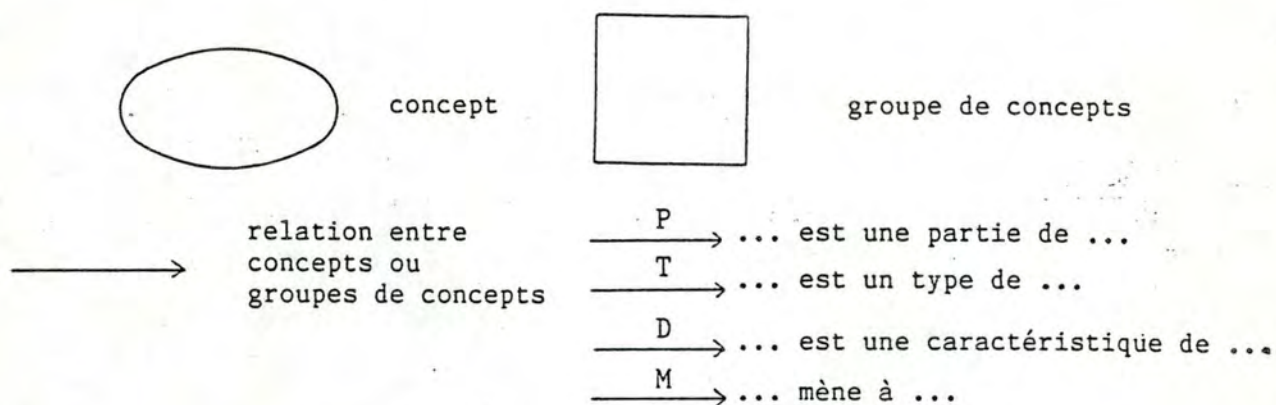
Luther trouvera une issue à son angoisse un jour qu'il relira un exposé de S. Paul sur la grâce : c'est Dieu qui, par pure libéralité et sans se soucier des mérites des hommes, accorde le salut à ceux qui s'en remettent à lui (ou en d'autres termes qui ont la foi);

Luther connaît une exaltation profonde à reconnaître ainsi le rôle tout-puissant et quasi exclusif de Dieu dans le salut et, en contre partie, à rabattre les prétentions dérisoires de l'homme.

Par ses doctrines, Luther rejoint un courant important de la théologie traditionnelle, qui insiste sur la grandeur de Dieu et la grâce plus que sur la dignité de l'homme et la valeur de ses oeuvres. On peut dès lors à juste titre contester l'originalité de la découverte de Luther. Ce qu'on ne peut contester, c'est qu'il a été le premier à exploiter cette découverte pour réformer fondamentalement la chrétienté.

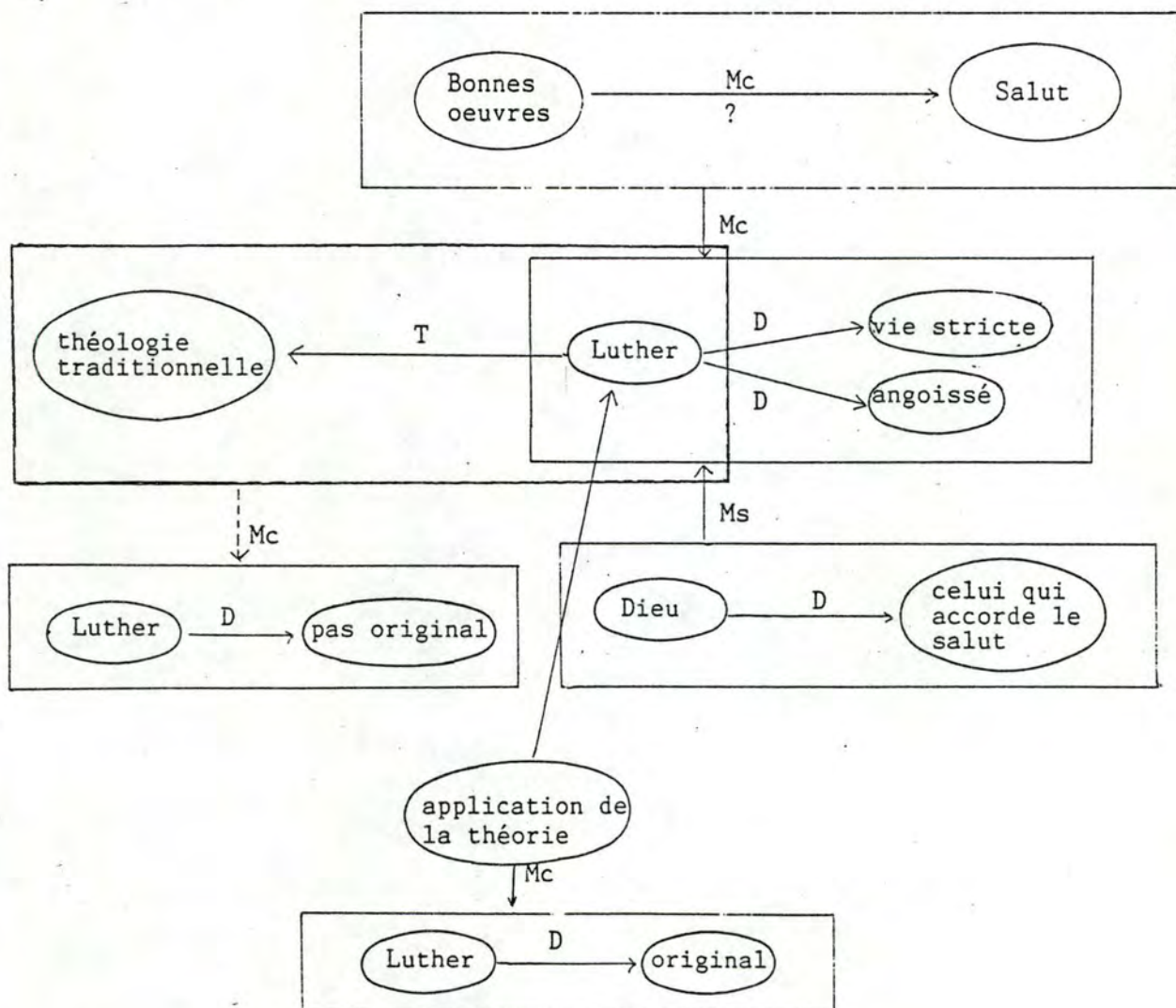
EXEMPLE DE TEST DE TYPE RÉSEAU.

1. Matériaux



2. Réponse:

Mc ... cause ...
 Ms ... solutions ...



EXEMPLE DE TEST DE TYPE LIAISON DES PARAGRAPHES D'UN TEXTE.

1. Matériaux

→ ... est ...	P → ... est une partie de ...
→ ... possède ...	T → ... est un type de ...
→ ... entraîne ...	D → ... est une caractéristique de ...
	M → ... mène à ...

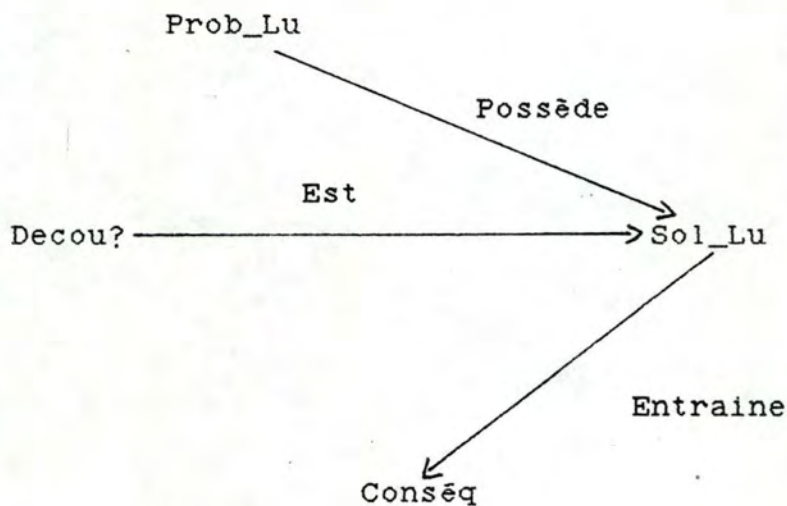
Mc ... cause ...

Ms ... solutions ...

Les identifiants des paragraphes introduits par une personne testée :

- pour le premier paragraphe : Prob_Lu
- pour le second paragraphe : Sol_Lu
- pour le troisième paragraphe : Conséq
- pour le quatrième paragraphe : Decou?

2. Réponse :



EXEMPLE DE TEST DE TYPE HIÉRARCHIE DES PARAGRAPHES D'UN TEXTE.

1. Matériaux

Les identifiants des paragraphes introduits par une personne testée :

- pour le premier paragraphe : Prob_Lu
- pour le second paragraphe : Sol_Lu
- pour le troisième paragraphe : Conséq
- pour le quatrième paragraphe : Decou?

2. Réponse :

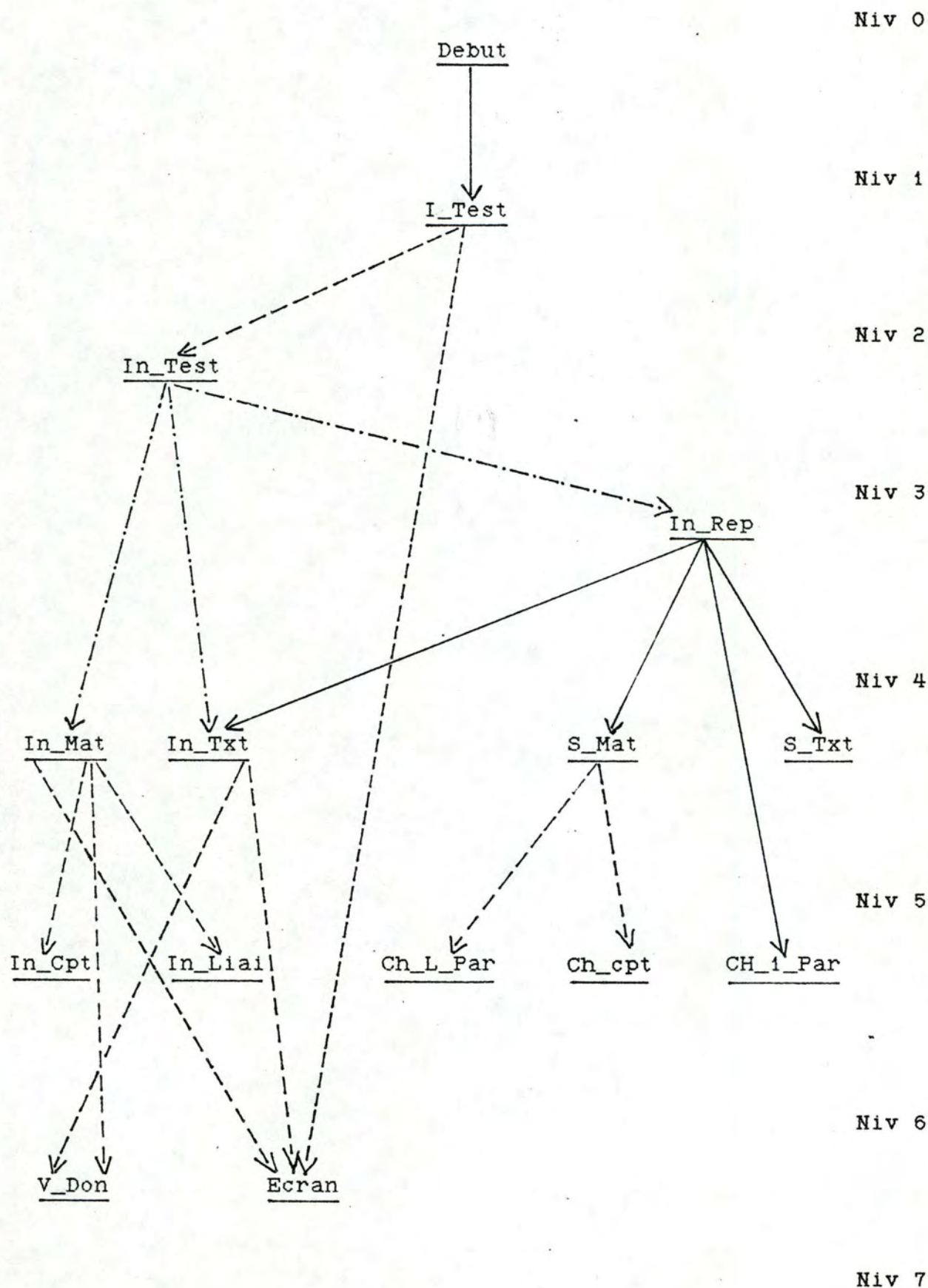
1. Prob_Lu

2. Sol_Lu

3. Conséq

3.1 Décou?

Annexe 2 : Graphes de la découpe en niveau



Niv 0

Debut

Niv 1

Réalisation

Niv 2

In_Rep_T

Niv 3

In_Rep

S_Rep

Niv 4

In_Texte

S_Mat

S_Texte

Niv 5

Ch_1_Par

Ch_L_Par

Ch_Cpt

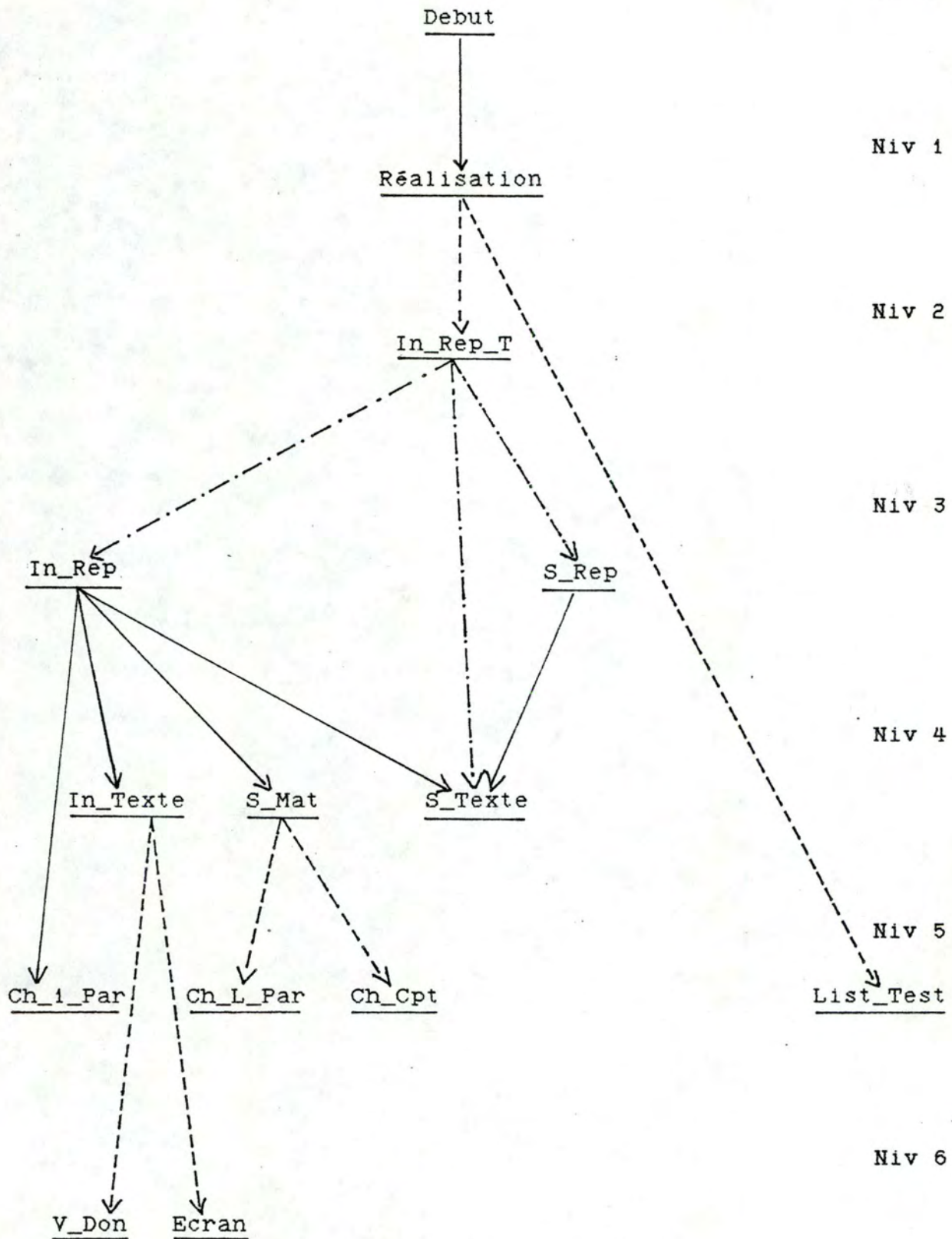
List_Test

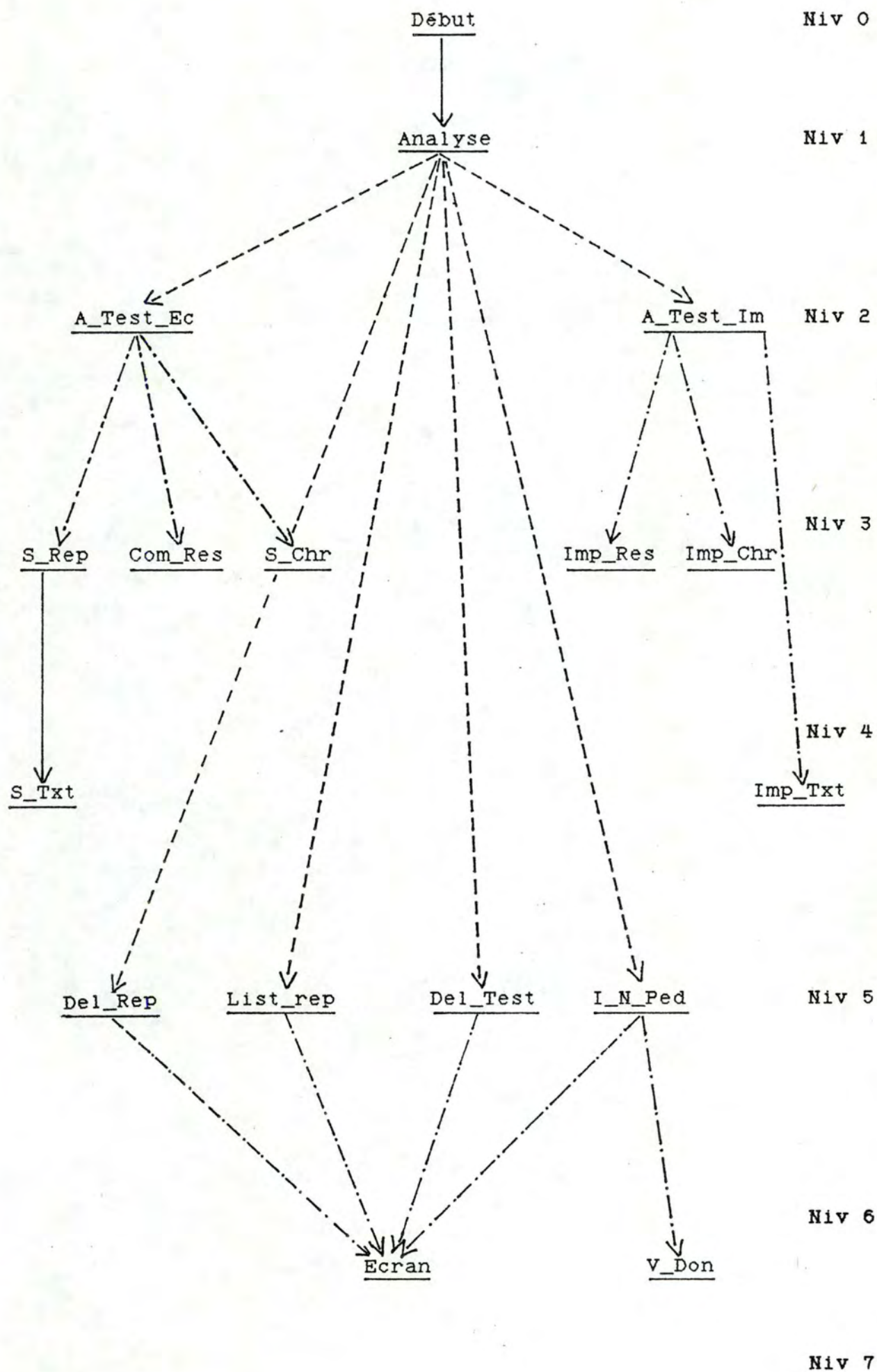
Niv 6

V_Don

Ecran

Niv 7





Annexe 3 : Présentation des écrans

INTRODUCTION

Le concepteur a choisi de regrouper les écrans en six groupes. Cette décision a été prise dans le but de diminuer la place nécessaire au module contenant les écrans. Cette décision est due au fait qu'un grand nombre d'éléments étaient communs à des écrans de même type.

Les types d'écrans sont déterminés par un numéro :

- écrans de type 1 :

Ce groupe contient les écrans qui présentent aux utilisateurs un menu. C'est-à-dire : qui offrent le choix entre plusieurs actions possibles.

- écrans de type 2 :

Ce groupe contient l'ensemble des écrans qui demandent aux utilisateurs d'introduire une ou plusieurs données dans le logiciel.

Ces écrans se différencient des autres types d'écrans par cette introduction de données par l'utilisateur. Ces données doivent être contrôlées en vue d'une acceptation ou d'un refus. Un contrôle de la longueur et de l'existence est effectué sur chaque donnée introduite en utilisant le module écran.

Un module spécial (Vérif_Don : niveau 6 de la découpe) vérifie la sémantique des données introduites.

- écrans de type 3 :

Ce groupe d'écrans contient ceux qui signalent à l'utilisateur que la(les) donnée(s) introduite(s) est(sont) refusée(s).

- écrans de type 4 :

Ce groupe contient les écrans qui demandent à l'utilisateur un choix du type 'oui' ou 'non'.

- écrans de type 5 :

Ce groupe contient les écrans qui indiquent à l'utilisateur à quel niveau il se trouve ou qui permettent à l'utilisateur de visionner une série de données sans choix à effectuer.

- écrans de type 6 :

Ce groupe contient les écrans qui présentent à l'utilisateur une liste de données dans laquelle il peut choisir un élément.

Les différents écrans qui sont présentés ici ne contiennent pas les écrans permettant de modifier les réponses, les textes, les matériaux introduits.

Si le système est agrandi, les concepteurs futurs pourront ajouter de nouveaux écrans dans les types d'écrans existant ou dans de nouveaux types selon les besoins.

Ecran numéro 0

------(type 1)

Utilisation :

Dans la procédure Debut.

Fonction :

Permet à l'utilisateur qui rentre dans le système de choisir ce qu'il veut faire parmi les possibilités qui lui sont présentées.

Contenu :

" Bonjour,
Parmi les quatres possibilités qui vous sont présentées, choisissez celle que vous voulez en vous servant de la souris."

- Introduire un test
- Réaliser un test
- Analyser un test ou détruire des éléments
- Quitter le programme

Ecran numéro 1

------(type 2)

Utilisation :

Dans la procédure Debut.

Fonction :

Si l'utilisateur désire introduire un test ou consulter des résultats, l'écran 2 lui demande d'introduire un nom de pédagogue.

L'utilisateur a droit à deux essais. Si au second, le nom donné n'existe pas, l'utilisateur est éjecté.

L'écran 2 vérifie que la longueur du nom introduit est ≥ 0 et inférieure à une longueur maximale donnée.

Contenu :

" Donnez le nom de pédagogue : "

Paramètres :

Erreur : boolean (* mis à true si le nom introduit est trop grand
mis à false si non *)

Ele1 : string[lid] (* contient le nom de pédagogue introduit *)

Exist1 : boolean (* mis à false si Erreur = true ou si l'élément
est de longueur nulle
mis à true si Erreur = false et lg-élé >0 *)

Ecran numéro 2

----- (type 2)

Utilisation :

Dans la procédure In_Nom_Ped.

Fonction :

Permet à l'utilisateur d'introduire un nouveau nom de pédagogue.
L'écran 2 vérifie si la longueur du nom introduit est ≥ 0 et inférieure à une longueur maximale.

Contenu :

" Donnez le nouveau nom de pédagogue : "

Paramètres :

Erreur : boolean (* mis à true si le nom introduit est trop grand
mis à false si non *)

Ele1 : string[lid] (* contient le nom de pédagogue introduit *)

Exist1 : boolean (* mis à false si Erreur = true ou si l'élément
est de longueur nulle
mis à true si Erreur = false et lg-élé >0 *)

Ecran numéro 3

------(type 1)

Utilisation :

Dans la procédure I_Test.

Fonction :

Si l'utilisateur a choisi la partie introduction d'un test, l'écran 3 lui est présenté pour qu'il puisse déterminer l'action qu'il veut réaliser.

Contenu :

" Choisissez, grâce à la souris entre :

- Introduire un nouveau test
- Retourner au menu principal

Paramètres :

Un nombre dont la valeur est :

- 1 si l'introduction est choisie,
- 2 si le retour au menu précédent est choisi;

Ecran numéro 4

------(type 4)

Utilisation :

Dans les procédures Choix-Test et I_Test.

Fonction :

Permet à l'utilisateur de déterminer s'il désire ou non consulter la liste des tests existants.

Contenu :

" Voulez-vous consulter la liste des tests existants ? "

" OUI " " NON "

Paramètres :

Un booléen ayant la valeur true si l'option oui est choisie, false si non.

Ecran numéro 5

------(type 6)

Utilisation :

Dans les procédures Choix_Test et I_Test.

Fonction :

Présenter, à l'écran, la liste des test existants et leurs types.

Contenu :

" Voici la liste des tests existants : "

T O T O

R é s e a u

T I T I

H i é r a r c h i e

" S U I T E "

Paramètre : /

Ecran numéro 6

------(type 2)

Utilisation :

Dans la procédure I_Test

Fonction :

Demande à l'utilisateur d'introduire le nom identifiant du test à introduire.

L'écran 6 vérifie si le nom introduit à une longueur ≥ 0 et inférieure à un maximum donné.

Contenu :

" Donnez le nom du test à introduire : "

Paramètres :

Erreur : boolean (* mis à true si le nom introduit est trop grand
mis à false si non *)

Ele1 : string[lid] (* contient le nom de pédagogue introduit *)

Exist1 : boolean (* mis à false si Erreur = true ou si l'élément
est de longueur nulle
mis à true si Erreur = false et $lg_ele > 0$ *)

Ecran numéro 7

------(type 3)

Utilisation :

Dans les procédures I_Test, I_Nom_Ped et In_Mat.

Fonction :

Message d'erreur indiquant que l'indentifiant introduit est déjà attribué. Demande l'introduction d'un autre identifiant.

Contenu :

" Ce nom n'est pas identifiant; il a déjà été attribué, introduisez en un autre.

O K "

Argument : /

Ecran numéro 8

------(type 2)

Utilisation :

Dans la procédure I_Test.

Fonction :

Demande à l'utilisateur d'introduire le type de test correspondant au non de test qui vient d'être introduit.

Si l'utilisateur introduit plusieurs caractères, l'écran ne tient compte que du premier introduit.

Contenu :

" Donnez le type du test :
soit r(éseau)
soit l(iaison)
soit h(iérarchie) "

Argument :

Le résultat de la fonction est un caractère.

Ecran numéro 9

------(type 3)

Utilisation :

Dans la procédure I_test.

Fonction :

Signale à l'utilisateur que le type de test introduit est incorrect.

Contenu :

" Le type de test introduit est incorrect. Introduisez en un autre.

O K

"

Argument : /

Ecran numéro 10

------(type 4)

Utilisation :

Dans la procédure In_Test.

Fonction :

Permet à l'utilisateur de signaler s'il veut ou non introduire un test explicatif.

Contenu :

" Voulez-vous introduire un test explicatif ? "

" OUI " " NON "

Argument :

Un booléen ayant la valeur true si l'option oui est choisie, false sinon.

Ecran numéro 11

------(type 1)

Utilisation :

Dans la procédure In_Test.

Fonction :

Permet à l'utilisateur de signaler qu'il veut ou ne veut pas introduire d'exemple de réponse.

Contenu :

" Voulez-vous introduire un exemple de réponse pour ce test ?
Si oui, choisissez si cet exemple doit apparaître aux
personnes testées progressivement ou non.

NON PROGRESSIVEMENT EN BLOC "

Argument :

Un nombre entier qui a comme valeur :

- 1 si l'option non est choisie,
- 2 si l'option progressivement est choisie,
- 3 si l'option en bloc est choisie.

Ecran numéro 12

------(type 5)

Utilisation :

Dans la procédure In_Test.

Fonction :

Signale à l'utilisateur-pédagogue qu'il va devoir introduire la
réponse-type correspondant au test en cours d'introduction.

Contenu :

" Vous allez introduire votre réponse-type.
Pour commencer, cliquer suite.

S U I T E "

Argument : /

Ecran numéro 13

------(type 5)

Utilisation :

Dans la procédure In_Test.

Fonction :

Signale à l'utilisateur qu'il va pouvoir introduire le texte de
travail du test en cours d'introduction.

Contenu :

" Vous allez introduire le texte de travail du nouveau test.
Si des mots ou caractères doivent être soulignés, utilisez
le caractère ^ au début et à la fin de ces éléments.
Pour commencer, cliquer suite.

S U I T E

"

Argument : /

Ecran numéro 14

------(type 2)

Utilisation :

Dans la procédure In_Texte dans le cas d'un texte de travail.

Fonction :

Demande à l'utilisateur-pédagogue d'introduire le titre de
l'en-tête du texte de travail, s'il existe.

Vérifie si la longueur du titre est ≥ 0 et l'existence du
titre.

Contenu :

" Donnez le contenu de l'en tête: (RETURN s'il ni en a pas)

"

Paramètres :

Erreur : boolean (* mis à true si le nom introduit est trop grand
mis à false si non *)

Ele2 : string[ltitr] (* contient l'en-tête introduit *)

Exist1 : boolean (* mis à false si Erreur = true ou si l'élément
est de longueur nulle
mis à true si Erreur = false et lg-élé > 0 *)

Ecran numéro 15

------(type 2)

Utilisation :

Dans la procédure In_Texte dans le cas d'un texte de travail.

Fonction :

Demande à l'utilisateur-pédagogue d'introduire le contenu de
l'en tête.

Contenu :

" Donnez le contenu de l'en tête en un seul bloc. (RETURN s'il n'y en a pas).
Employez la touche RETURN pour indiquer la fin de l'introduction.

Paramètre :

Ele2 : string[1titr] (* contient l'en-tête introduit *)

Ecran numéro 16

------(type 3)

Utilisation :

Dans la procédure In_Texte, dans le cas d'un texte de travail;

Fonction :

Signale à l'utilisateur que l'en tête introduit possède un titre mais pas de contenu. Ceci n'étant pas admis un contenu doit être introduit ou le titre de l'en tête supprimé.

Contenu :

" L'en-tête introduit à un titre et pas de contenu. Ceci n'étant pas accepté. Introduisez le contenu de l'en tête ou supprimez là en tapant simplement RETURN.

OK "

Argument : /

Ecran numéro 17

------(type 2) .

Utilisation :

Dans la procédure In_Texte dans la cas d'un texte de travail.

Fonction :

Demande à l'utilisateur d'introduire le titre du paragraphe suivant.

Contenu :

" Donnez le titre du paragraphe suivant : "

Paramètres :

Erreur : boolean (* mis à true si le nom introduit est trop grand
mis à false si non *)

Ele2 : string[1titr] (* contient l'en-tête introduit *)

Exist1 : boolean (* mis à false si Erreur = true ou si l'élément
est de longueur nulle
mis à true si Erreur = false et lg-élé >0 *)

Ecran numéro 18

------(type 2)

Utilisation :

Dans la procédure In_Texte.

Fonction :

Demande à l'utilisateur d'introduire le contenu d'un paragraphe
de texte.

Contenu :

" Donnez le contenu du paragraphe suivant, en un bloc (RETURN
si l'introduction est terminée).
C O N T E N U : "

Paramètre :

Exist2 : boolean (* s'il n'y a pas de contenu Exist2 a la valeur
false, true si non *)

Ecran numéro 19

------(type 3)

Utilisation :

Dans la procédure In_Texte.

Fonction :

Signale à l'utilisateur qu'il a introduit un paragraphe avec un
titre et pas de texte. Comme ceci n'est pas admis, le contenu
du paragraphe doit être introduit ou le titre supprimé.

Contenu :

" Vous avez introduit un parargraphe vide avec un titre. Ceci
n'est pas accepté. Introduisez un contenu de paragraphe ou
tapez return pour éliminer le titre introduit. "

Argument : /

Ecran numéro 20

------(type 5)

Utilisation :

Dans la procédure In_Test.

Fonction :

Signale à l'utilisateur qu'il va pouvoir introduire les matériaux correspondants au test en cours d'introduction.

Contenu :

" Vous allez introduire les matériaux du test en cours d'introduction.

SUITE

Argument : /

Ecran numéro 21

------(type 4)

Utilisation :

Dans la procédure In_Mat, dans le cas de tests de type réseau.

Fonction :

Demande à l'utilisateur s'il veut permettre utilisation de la liaison rectangle dans le test en cours d'introduction.

Contenu :

" Pour ce test, voulez-vous permettre l'utilisation de la liaison rectangle ?

" O U I "

" N O N "

Argument :

un booléen ayant la valeur true si l'option oui est choisie, false sinon.

----- (type 2)

Dans la procédure In_MAT.

Demander l'introduction de la suite de caractères contenant la signification en langue naturelle qui identifie un concept.

" Introduisez la signification en langue naturelle identifiant le
c o n c e p t s u i v a n t : "

```
Exist1 : boolean (* mis à false si Erreur = true ou si l'élément
                    est de longueur nulle
                    mis à true si Erreur = false et lg-élé >0 *)
```

------(type 3)

Dans les procédures In_Mat, In_Texte, Réalisation

Avertir l'utilisateur que l'élément introduit n'est pas identifiant et demander l'introduction d'un autre élément.

" Cet élément n'est pas identifiant, veuillez en réintroduire un autre.

OK H

Argument : /

Ecran numéro 24

------(type 2)

Utilisation :

Dans la procédure In_Mat.

Fonction :

Demande d'introduire l'identifiant mnémotechnique d'un concept ou une liaison flèche.

Contenu :

" Veuillez introduire l'identifiant mnémotechnique de l'élément "

Argument :

Erreur : boolean (* mis à true si le nom introduit est trop grand
mis à false si non *)

Ele1 : string[lid] (* contient l'identifiant court du concept
introduit *)

Exist2 : boolean (* mis à false si Erreur = true ou si l'élément
est de longueur nulle
mis à true si Erreur = false et lg-élé >0 *)

Ecran numéro 25

------(type 3)

Utilisation :

Dans les procédures In_Texte, In-Mat, In_Test.

Fonction :

Signale à l'utilisateur que la donnée introduite est trop longue par rapport à la limite permise.

Contenu :

" l'élément introduit est trop long. La longueur maximum pour :
un identifiant mnémotechnique est : 6 carac
un titre ou un identifiant en langage naturel est : 20 carac "

Argument : /

Ecran numéro 26

------(type 6)

Utilisation :

Dans la procédure Del_In_Test.

Fonction :

Permet à l'utilisateur de visionner et de choisir parmi les tests existants celui qui va être supprimé ou d'annuler sa commande.

Contenu :

" Choisissez l'identifiant du test que vous désirez supprimer."

TITI Liaison

TOTO Réseau

ANNULATION

Argument :

Un nombre dont la valeur correspond

- l'élément qui a été choisi, s'il existe
- à une valeur correspondant à l'annulation de la commande.

Ecran numéro 27

------(type 4)

Utilisation :

Dans les procédures Del_Test, Del-Rep, In_Nom_Ped.

Fonction :

Demande de confirmation de la suppression de l'élément choisi

Contenu :

" Confirmez-vous la suppression de l'élément ? "

" O U I "

" N O N "

Argument :

Un booléen dont la valeur est true si c'est l'option oui qui est choisie, false si non.

Ecran numéro 28

------(type 6)

Utilisation :

Dans le module Realisation.

Fonction :

Demande à l'utilisateur de choisir parmi les tests existants celui qu'il va réaliser.

Contenu :

" Voici la liste des tests existants et leur type. Choisissez le test que vous voulez réaliser ou retourner au menu général.

TOTO	Réseau
TITI	Liaison

MENU GENERAL

Argument :

Un nombre dont la valeur correspond à

- l'élément qui a été choisi, s'il existe
- une valeur correspondant à l'annulation de la commande.

Ecran numéro 29

------(type 2)

Utilisation :

Dans la procédure Realisation.

Fonction :

Demande à l'utilisateur d'introduire son nom identifiant.
Vérifie la longueur de l'élément introduit.

Contenu :

" Donnez votre nom en tant que personne testée : "
(6 car maximum)

Paramètres :

Erreur : boolean (* mis à true si le nom introduit est trop grand
mis à false si non *)

Ele1 : string[lid] (* contient le nom introduit *)

Exist1 : boolean (* mis à false si Erreur = true ou si l'élément
est de longueur nulle
mis à true si Erreur = false et lg-élé >0 *)

Ecran numéro 30

------(type 5)

Utilisation :

Dans la procédure Réalisation.

Fonction :

Explique à l'utilisateur-testé dans quel ordre les différents éléments du test à réaliser vont apparaître.

Contenu :

" Les éléments qui constituent le test vont apparaître de la façon suivante :

- (- présentation d'un texte explicatif)
- (- présentation d'un exemple)
- la présentation du texte de travail
- introduction de la réponse
- présentation de la réponse-type.

S U I T E

"

Arguments :/

Ecran numéro 31

------(type 5)

Utilisation :

Dans la procédure In_Rep_Test.

Fonction :

Annoncer la présentation du texte explicatif avec explication des commandes utilisables.

Contenu :

" Le texte explicatif va être présenté.

Vous pouvez utiliser les commandes suivantes :

- accès à la page précédente -> cliquer p.Avant
- accès à la page suivante -> " p.Après
- fin de la consultation -> " arrêt

S U I T E

"

Argument : /

Ecran numéro 32

------(type 5)

Utilisation :

Dans la procédure In_Rep_Test.

Fonction :

Présenter à l'utilisateur un exemple de réponse au type de test réalisé.

Déterminer les commandes qui vont pouvoir être utilisées par l'utilisateur-testé dans le cas d'une présentation progressive.

Contenu :

" Un exemple de réponse va vous être présenté.

Les commandes utilisables sont :

- reprise du défilement -> cliquer OK

- fin de la consultation avant fin exemple -> cliquer Fin

S U I T E

"

Argument : /

Ecran numéro 33

------(type 5)

Utilisation :

Dans la procédure In_Rep_Test.

Fonction :

Annoncer la présentation du texte de travail ainsi que des commandes utilisables sur ce texte.

Contenu :

" Le texte de travail va vous être présenté. Après cette première consultation, vous pourrez réaliser votre réponse.

Les commandes utilisables sur le texte sont :

- page précédente -> cliquer p.Avant

- page suivante -> " p.Après

- fin consultation -> " arrêt

S U I T E

"

Argument : /

Ecran numéro 34

------(type 5)

Utilisation :

Dans procédure In_Rep_Test.

Fonction :

Indiquer à l'utilisateur qu'il va introduire sa réponse et les commandes utilisables pour accéder au texte de travail et aux matériaux dans le cas de tests 'réseau' ou 'liaison'.

Contenu :

"Vous allez introduire votre réponse.

Les commandes disponibles sont :

- accès au texte de travail -> cliquer txt-t
- demande de création d'un texte explicatif -> cliquer txt-e
- fin de la réalisation de la réponse -> cliquer fin

(- accès aux concepts -> cliquer cpts)

(- accès aux liaisons -> cliquer liais)

S U I T E

"

Argument : /

Ecran numéro 35

------(type 5)

Utilisation :

Dans la procédure S_Rep.

Fonction :

Présentation de la réponse type à l'utilisateur.

Contenu :

" La réponse-type du pédagogue va être présentée.

Les commandes utilisables sont :

- reprise du défilement -> cliquer Go
- fin de la consultation -> " fin

S U I T E

"

Argument : /

Ecran numéro 36

------(type 5)

Utilisation :

Dans le module Realisation

Fonction :

Signaler à l'utilisateur que le test est terminé et qu'il retourne au niveau du menu général.

Contenu :

" Au revoir,
le test est terminé et vous retournez au niveau du
premier menu !

S U I T E

"

Argument : /

Ecran numéro 37

------(type 1)

Utilisation :

Dans la procédure Analyse.

Fonction :

Demander à la personne qui veut consulter des résultats de tests quel type d'opérations il veut réaliser.

Contenu :

" Choisissez, grâce à la souris, l'action que vous voulez réaliser sur des résultats de tests.

- Imprimer sur papier ou à l'écran une réponse.
- Détruire une(des) réponse(s) à un(des) test(s).
- Détruire, introduire un nom de pédagogue
- Détruire un test
- Consulter la liste des tests existants.
- Retourner au menu général.

"

Argument :

Un nombre dont la valeur est :

- 1 si l'impression est choisie,
- 2 si la destruction de réponse est choisie
- 3 si la destruction-intro d'un nom de pédagogue
- 4 si la destruction d'un test est choisie
- 5 la consultation de la liste des tests existants est choisie
- 6 retour au menu

Ecran numéro 38

------(type 4)

Utilisation :

Dans le module Analyse.

Fonction :

Si l'utilisateur a choisi d'imprimer un résultat, on lui demande si cela doit être réalisé à l'écran ou sur papier.

Contenu :

" Voulez vous que la réponse soit imprimée :

• "SUR PAPIER" "A L'ECRAN"

"

Arguments :

un booléen ayant la valeur true si l'option sur papier est choisie, false sinon.

Ecran numéro 39

------(type 6)

Utilisation :

Dans la procédure Analyse.

Fonction :

Si l'utilisateur a choisi l'impression d'un texte, l'écran 29 lui présente les tests existants pour qu'il puisse choisir celui qui l'intéresse.

Contenu :

" Choisissez le test à imprimer :

TOTO	Réseau
TITI	Liaison

"

Argument :

Un nombre qui identifie l'élément choisi ou l'option d'annulation de la commande.

Ecran numéro 40

----- (type 6)

Utilisation :

Dans le module Analyse.

Fonction :

Présenter à l'utilisateur les personnes existantes qui ont réalisé le test choisi.

Contenu :

" Voici les personnes ayant réalisé le test choisi. Choisissez la personne à traiter.

DUPOND
DURAND

"

Argument :

Un nombre qui identifie l'élément choisi ou l'option d'annulation de la commande.

Ecran numéro 41

----- (type 1)

Utilisation :

Dans les procédures S_Rep et In_Rep.

Fonction :

Si l'utilisateur a choisi d'imprimer sur papier un test de type réseau d'une personne déterminée, l'écran 31 lui demande de choisir entre différents éléments qui peuvent être imprimés.

Contenu :

" Désirez vous imprimer :

- La liste chronologique des matériaux choisis;
- Le réseau réalisé.
- Le texte introduit par le testé, s'il existe.
- La comparaison entre deux réseaux du même test.

Désirez vous :

- Retourner au menu précédent. "

Argument :

un nombre dont la valeur est :

- 1 si l'option liste chronologique a été choisie,
- 2 si l'option réseau a été choisie
- 3 si l'option texte a été choisie
- 4 si l'option comparaison a été choisie
- 5 si l'option retour au menu à été choisie

Ecran numéro 42

------(type 6)

Utilisation :

Fonction :

Si l'utilisateur a choisi d'imprimer une comparaison entre deux réseaux, il doit choisir la seconde réponse qui doit être comparée avec la réponse qui vient d'être choisie.

Contenu :

" Voici la liste des personnes ayant effectué le même test que la première personne choisie. Choisissez la personne avec laquelle la comparaison va s'effectuer.

DURAND

Argument :

Un nombre qui identifie l'élément choisi ou l'option d'annulation de la commande.

Ecran numéro 43

------(type 5)

Utilisation :

Fonction :

Expliquer, à la personne qui veut effectuer la comparaison entre deux réseaux, dans quels types de caractères apparaissent les différentes parties.

Contenu :

- " - Les éléments communs aux deux réponses apparaissent en caractères gras;
- Les éléments appartenants à la première personne apparaissent en caractères italiques.
- Les éléments appartenants à la seconde personne apparaissent en caractères normaux. "

SUITE

Argument : /

Ecran numéro 44

------(type 1)

Utilisation :

Fonction :

Si le test à imprimer est du type hiérarchie ou liaison, l'écran 44 demande à l'utilisateur quelle opération il désire effectuer sur la réponse choisie.

Contenu :

- " Désirez vous :
 - imprimer Le texte de travail muni des identifiants choisis;
 - imprimer La réponse réalisée;
 - Retourner au menu précédent. "

Argument :

Un nombre dont la valeur est :

- 1 si l'option impression du texte a été choisie
- 2 si l'option impression de la réponse a été choisie
- 3 si l'option retour au menu précédent a été choisie

Ecran numéro 45

------(type 5)

Utilisation :

Fonction :

L'écran 45 présente à l'utilisateur les manipulations possibles sur un texte.

Contenu :

" Les manipulations possibles sur un texte sont :

- Page avant -> cliquer p.Avant ;
- Page arrière -> " p.Apres ;
- quitter le texte -> " arret.

S U I T E

"

Argument : /

Ecran numéro 46

------(type 2)

Utilisation :

Fonction :

Demander à l'utilisateur d'introduire la signification en langue naturelle de la liaison suivante.

Contenu :

" Introduisez la signification en langue naturelle de la liaison suivante. (return s' il n'existe pas)

"

Arguments :

Erreur : boolean (* mis à false si la suite introduite est trop longue *)

Ele2 : string[Ltitr] (* contient la suite introduite *)

Exist1 : boolean (* mis à false si Erreur est à true ou si la longueur de l'élément introduit = 0, mis à true si non *)

Ecran numéro 47

------(type 3)

Utilisation :

Dans la procédure In_test

Fonction :

Signaler à l'utilisateur que le nom de pédagogue introduit n'existe pas et qu'il a droit à deux essais.

Contenu :

" Le nom introduit n'existe pas. Attention vous avez droit à 2
e s s a i s . "

SUITE

Argument : /

Ecran numéro 48

------(type 6)

Utilisation :

Dans la procédure In_Nom_Ped.

Fonction :

Présenter à l'utilisateur la liste des pédagogues existants pour permettre le choix d'un pédagogue à supprimer. Si aucun pédagogue n'est choisi, l'utilisateur clique l'ANNULATION de cet écran.

Contenu :

" Voici la liste des pédagogues existants. Choisissez celui qui doit être supprimé.

A N N U L A T I O N

"

Arguments :

Un nombre qui identifie l'élément choisi ou l'option d'annulation de la commande.

Ecran numéro 49

------(type 3)

Utilisation :

Dans les procédures In_Texte et In_Mat.

Fonction :

Indiquer à l'utilisateur que la signification qui a été introduite n'a pas reçu d'identifiant. Ce qui est incorrect.

Contenu :

" Vous n'avez pas introduit d'identifiant pour cette signification. Veuillez le faire.

OK "

Argument : /

Ecran numéro 50

------(type 6)

Utilisation :

Dans les procédures Del_Test et Del_Rep.

Fonction :

Présenter à l'utilisateur la liste des tests existants parmi lesquels, il pourra en choisir un ou annuler sa commande.

Contenu :

" Voici la liste des tests existants. "

Argument :

Un nombre qui identifie l'élément choisi ou l'option d'annulation de la commande.

Ecran numéro 51

------(type 4)

Utilisation :

Dans la procédure Del-Rep

Fonction :

Si l'utilisateur désire détruire des réponses, l'écran 51 lui demande de choisir entre la destruction d'une seule réponse que l'utilisateur devra par la suite désigner ou la destruction de toutes les réponses existantes au test choisi.

Contenu :

" Voulez vous détruire toutes ou une seule réponse(s) à ce test ?

"T O U T E S"

"U N E"

Argument :

Un booléen dont la valeur est true si l'option toutes est choisie, false sinon.

Ecran numéro 52

------(type 6)

Utilisation :

Dans la procédure Del_Rep.

Fonction :

Si l'utilisateur a choisi précédemment de détruire une seule réponse d'un test donné, l'écran 52 lui permet de désigner cette réponse.

Contenu :

" Voici la liste des personnes ayant réalisés ce test. Choisissez celle dont vous voulez détruire la réponse.

DUPOND

DURAND

Argument :

Un nombre qui identifie l'élément choisi ou l'option d'annulation de la commande.

Ecran numéro 53

------(type 5)

Utilisation :

Dans la procédure Analyse.

Fonction :

Permet à l'utilisateur de choisir entre une présentation des tests existants dans l'ordre alphabétique avec pour chacun les personnes l'ayant effectué ou une présentation des personnes testées existantes par ordre alphabétique avec les tests effectués par chacune.

Contenu :

" Voulez vous consulter :

- La liste des tests par ordre alphabétique avec pour chacun la liste alphabétique des personnes ayant effectué le test.
- La liste alphabétique des personnes avec pour chacune la liste alphabétique des tests effectués.
- la liste alphabétique des tests réalisés par une personne déterminée.
- la liste alphabétique des personnes ayant réalisé un test déterminé.
- retour au menu précédent

Argument :

un nombre dont la valeur est :

- 1 si la première liste est choisie,
- 2 " " seconde " " "
- 3 " " troisième " " "
- 4 " " quatrième " " "
- 5 si le retour au menu précédent est choisi.

Annexe 4 : Présentation des primitives d'accès

4.1 COLLECTION DES TESTS EXISTANTS

* Aj_Test (In_Test: string[Lid], T_Test : integer, N_Test
: integer, N_Ped : integer)

Préconditions :

- Il n'existe pas de test appartenant à la liste des tests existants dont l'intitulé est égal à In_Test.
- le type du test à introduire est soit '1' soit '2' soit '3'. (réseau, liaison ou hiérarchie)
- N_Ped est un numéro interne identifiant un pédagogue existant.
- Comp détermine l'existence ou l'absence d'éléments facultatifs.
- N_Test est un numéro non encore attribué à un test existant.

Fonction :

Ajoute un nouvel élément à la collection des tests existants et attribue à cet élément un numéro identifiant.

L'élément ajouté est formé d'un intitulé identifiant, d'un type de test, d'un identifiant interne d'un numéro de pédagogue l'ayant créé et d'un numéro identifiant.

Postconditions :

- un numéro identifiant a bien été attribué à l'élément introduit.
- l'élément introduit a bien été enregistré.

* Del_Test (N_Test : integer)

Précondition :

- le test dont le numéro est N_Test existe.

Fonction :

- rend l'élément appartenant à la liste des tests existants et correspondant au numéro N_Test inaccessible.

Postconditions :

- Le test dont le numéro est N_Test n'appartient plus à la liste des tests existants.

```
* Exist_test ( In_Test: string[Lid] ; N_Test : integer;  
               var Exist, Fin_f : boolean );
```

Précondition :

- soit on donne le numéro du test (N_Test >= 0) et l'intitulé In_Test est vierge.
soit on donne l'intitulé du test (In_Test <> ' ') et le numéro du test N_Test est < 0.

Fonction :

- Vérifie que le test dont on a donné le numéro ou l'intitulé existe.

Postconditions :

- Si le test existe, Exist est mis à vrai, Fin_f à faux.
- Si le test n'existe pas, Exist est mis à faux et si le nom ou le numéro n'appartient pas au fichier, Fin_f est mis à vrai. Si non, Fin_f est mis à faux.
(permet de déterminer si l'enregistrement visé est dans le fichier mais à été supprimé ou s'il n'appartient pas du tout au fichier.)

```
* Num_Test ( In_Test : string[Lid] ; var N_Test : integer)
```

Précondition :

- L'intitulé In_Test correspond à un test existant.

Fonction :

- Recherche le numéro attribué à l'intitulé In_Test donné.

Postcondition :

- Num_Test a le numéro attribué au test dont l'intitulé est In_Test.


```
* Lir_Test ( N_Test : integer ; var In_Test : string[Lid];
             var T_Test : integer ; var N_Ped : integer;
             var Comp : boolean );
```

Précondition :

- Le numéro Num_Test correspond à un test existant.

Fonction :

- Recherche et fourni l'intitulé, le type, l'auteur et la composition du test correspondant au numéro Num_Test fourni.

Postconditions :

- In_Test contient l'intitulé correspondant au numéro N_test.
- N_Ped contient le numéro de son auteur
- T_Test contient le type du test
- Comp détermine l'existence ou l'absence des deux éléments facultatifs d'un test.

4.2 COLLECTIONS DES TABLES DE CORRESPONDANCE

```
* Aj_Mat ( Sign_Mat : string[Ltitr]; Id_Mat : string[Lid];
           N_Test : integer ; N_Mat : integer);
```

Préconditions :

- N_Test est un numéro interne identifiant attribué à un test existant.
- Sign_Mat et Id_Mat sont <> ' ' et identifient chacun un élément qui n'appartient pas à la collection des matériaux existants.
- N_Mat est le numéro de matériaux qui sera attribué à l'élément qui va être introduit.

Fonction :

Ajoute un nouveau élément dans le fichier des matériaux existants. Cet élément est un matériau formé d'un signification Sing_Mat, d'un identifiant Id_Mat et d'un numéro N_Mat.

Postcondition :

- Un nouvel élément a été introduit dans la collection des matériaux du test dont le num. est N_Test.

```
* Del_Mat (N_Mat : integer; Id_Mat : string[Lid]; N_Test : integer)
```

Préconditions :

- N_Test est un numéro interne identifiant attribué à un test existant.
- N_Mat est un numéro interne identifiant attribué à un matériau existant et appartenant au test traité N_Test.
- Id_Mat est un intitulé court identifiant attribué à un matériau existant et appartenant au test traité.
- soit N_Mat >= 0 et Id_Mat = ' ' ,
soit N_Mat < 0 et Id_Mat <> ' ' ,

Fonction :

Elimine du fichier contenant les matériaux du test, identifié par N_Test, le matériau identifié par le contenu de N_Mat ou de Id_mat.

Postcondition :

- l'élément correspondant à N_Mat ou Id_Mat ne fait plus partie des matériaux existants pour ce test.

```
* Exist_Mat (Id_Mat : string[Lid]; N_Test, N_Mat : integer ;  
var Exist, Fin_f : boolean);
```

Préconditions :

- N_Test est un numéro interne identifiant attribué à un test existant.
- Id_Mat est un intitulé mnémotechnique identifiant.
- N_Mat est un entier.
- soit Id_Mat <> ' ' et N_Mat < 0.
soit Id_Mat = ' ' et N_Mat >= 0.

Fonction :

Vérifie si le matériau dont on a donné soit l'intitulé mnémotechnique Id_Mat soit le numéro interne N_Mat existe dans la liste des matériaux utilisable pour le test identifié par N_Test.

Postcondition :

Si un matériau existant correspond à l'identifiant introduit, Exist = true et Fin_f = false
sinon Exist = false et si le matériau donné n'appartient pas au fichier Fin_f = true, si non Fin_f = false.

```
* Num_Mat (Id_Mat : string[Lid]; N_Test : integer; var N_Mat : integer)
```

Préconditions :

- N_Test est un numéro interne identifiant attribué à un test existant.
- Id_Mat est un identifiant mnémotechnique attribué à un matériau existant.

Fonction :

Recherche le numéro attribué au matériau dont l'intitulé est Id_Mat.

Postcondition :

- N_Mat contient le numéro interne identifiant le matériau correspondant à Id_Mat.

```
* Lir_Mat (var Id_Mat : string[Lid]; var Sign_Mat : string[Ltitr]; N_Mat, N_Test : integer);
```

Préconditions :

- N_Test est un numéro interne identifiant attribué à un test existant.
- N_Mat est un numéro interne identifiant attribué à un matériau existant.

Fonction :

Fournir les identifiants en langue naturelle et mnémotechnique (Sign_Mat et Id-Mat) du matériau existant correspondant au numéro donné N_Mat pour le test identifié par N_Test.

Postcondition :

Id_Mat et Sign_Mat contiennent respectivement les identifiants en langue naturelle et mnémotechnique associés au matériau identifié par le numéro interne N_Mat.

* Ecrire_Nbcon (var N_Test, Nbcon : integer);

Précondition :

- N_Test est un numéro identifiant un test existant dont le type est réseau de concepts.

Fonction :

Ecrire dans le fichier F_Nbcon, un nouvel enregistrement contenant le nombre de concepts introduits Nbcon et le numéro de test N_Test pour lequel ces concepts ont été introduits.

Postcondition :

- Le nouvel élément a bien été enregistré.

* Lire_Nbcon (N_Test : integer; var Nbcon : integer);

Précondition :

- N_Test est un numéro attribué à un test existant.

Fonction :

Lire le nombre de concepts introduits pour le test dont le numéro interne est N_Test.

Postcondition :

- Nbcon contient le nombre de concepts introduits pour le test dont le numéro est N_Test.

4.3 COLLECTIONS DES RÉPONSES DES TESTS DE TYPE LIAISON.

* Aj_Id_O (N_Test, N_Pers, N_§, N_Text : integer; Id_§ : string[Lid]; Typ-R : char);

Préconditions :

- N_Test est un numéro appartenant à un test existant.
- N_Pers est un numéro identifiant une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, test ou exemple.

Fonction :

Introduit un élément supplémentaire (formé d'un identifiant de paragraphe origine, le numéro du paragraphe et le numéro de texte associé, s'il existe) dans le fichier correspondant à la personne et au test donnés.

Postcondition :

- L'élément introduit fait bien partie de la réponse.

```
* Aj_liais ( N_Test, N_Pers, N_Rel, N_Txt, N_C, N_O : integer;  
              Typ_R : char);
```

Précondition :

- N_Test, N_Pers, N_Rel, N_O, N_C sont des numéros qui existent.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Ajoute dans le fichier réponse, correspondant aux numéros de test et de personne donnés, l'information suivante :
L'identifiant de paragraphe dont le numéro est N_O est l'origine d'une relation dont le numéro est N_Rel, vers un identifiant de paragraphe cible dont le numéro est N_C. Un texte explicatif peut être associé à cette introduction si N_Test <> 0.

Postcondition :

- Le fichier réponse contient l'information définie dans la fonction de la primitive.

```
* Del_Id ( N_Test, N_Pers, N_Par : integer; Typ_R : char);
```

Préconditions :

- le numéro N_Test est un numéro de test existant
- le numéro N_Pers est un numéro de personne testée existant
- N_Par est un numéro identifiant qui a été attribué à un des paragraphes du texte de travail.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Rendre, dans le fichier réponse correspondant au numéro de test N_Test et à la personne N_Pers, inaccessible les records ayant comme numéro de paragraphe d'origine N_Par (élimine le record et les liaisons partant de celui-ci). Eliminer le(s) élément(s) de record(s) contenant N_Par comme identifiant de paragraphe cible (élimine la(les) liaison(s) arrivant à N_Par). Eliminer le texte associé à un élément supprimé, s'il existe.

Postconditions :

- Le numéro identifiant N_Par ne fait plus partie de la réponse traitée et il n'existe plus de liaison partant ou arrivant à N-Par dans la réponse.
- Il n'existe plus de texte associés aux liaisons supprimées ou à N_Par.

```
* Del_Liais ( N_Test, N_Pers, N_Liais, N_O, N_C : integer ;  
                                     Typ_R : char );
```

Préconditions :

- Le numéro N_Test est un numéro attribué à un test existant.
- Le numéro N_Pers est un numéro attribué à une personne existante.
- Le numéro N_Liais est le numéro d'une liaison existant entre deux paragraphes du texte de travail du test concerné dont les numéros identifiants sont respectivement N_O et N_C.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Elimine, dans le fichier réponse correspondant aux numéros de test et de personne donnés (N_Pers et N_Test), la liaison dont le numéro identifiant est N_Liais et qui relie les deux paragraphes dont les numéros sont N_O et N_C.

Elimination des informations se rapportant à cette liaison. C-à-d : l'indication du paragraphe cible et le texte explicatif, s'il existe.

Postconditions :

- La liaison N_Liais n'existe plus entre les identifiants Id_O et Id_C.
- Les informations existantes sur cette liaison ont été supprimées.

```
* Exist-N§ ( N_Test, N_Pers, N_§ : integer; var Exist, Fin_f  
            : boolean; var Id_§ : string; Typ_R : char );
```

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne testée existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Détermine si le numéro de paragraphe donné appartient à la réponse déterminée par N_Test et N_Pers et si oui, fourni son identifiant.

Postcondition :

- Si le paragraphe dont le numéro est N_§ existe dans la réponse, Exist = true, Fin_f = false, Id_§ contient son identifiant.
Sinon Exist = false, Id_§ est indéterminé et si le paragraphe n'appartient pas à la réponse, Fin_f = true.

```
* Lir_IdO (N_Test, N_Pers : integer; Typ_R :char; F_IdO :string);
```

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne testée existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.
- F_IdO contient le nom du fichier dans lequel les intitulés de paragraphes origines seront placés.

Fonction :

Fournir la liste des identifiants (F_IdO) de paragraphes origine (Id_O), le numéro de paragraphe du texte de travail correspondant (N_§) et le numéro de texte explicatif, s'il existe.

Postcondition :

F_IdO contient les identifiants de paragraphes origines, les numéros de paragraphes correspondants et le numéro de texte explicatif, s'il existe.

```
* Lir_Liais ( N_Test, N_Pers, N_§ : integer; Typ_R : char;  
F_IdC : string);
```

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Le numéro N_§ est un numéro identifiant attribué à un paragraphe existant.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.
- F_IdC contient le nom du fichier qui contiendra les numéros de paragraphes cibles et les liaisons flèches employées à partir du paragraphe N_§.

Fonction :

Lire, dans le fichier réponse correspondant, les numéros de liaisons dont le paragraphe ayant le numéro N_§ est l'origine, les numéros paragraphes cibles et le texte explicatif associé, s'il existe. Placer les éléments lus dans le fichier F_IdC.

Postcondition :

- Le fichier F_IdC contient l'ensemble des numéros de liaisons partant de N_§ et le numéro de paragraphe cible correspondant ainsi que les numéros de textes associés à ces liaisons, s'ils existent.

4.4 COLLECTIONS DES RÉPONSES DE TYPE HIÉRARCHIE

* Lir_Id§ (N_Test, N_Pers, N_§ : integer; var Id_§ : string[Lid]
Typ_R : char; var Hier : string);

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Le numéro N_§ est un numéro identifiant attribué à un paragraphe existant.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Fournir l'identifiant du paragraphe (Id_§) dont le numéro est N_§ ainsi que sa hiérarchie Hier.

Postcondition :

- La variable Id_§ contient l'identifiant du paragraphe dont le numéro est N_§ et la variable Hier contient sa hiérarchie.

* Exist_N§ (N_Test, N_Pers, N_§ : integer; var Exist, Fin_f :
boolean ; Typ_R : char);

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Détermine si le paragraphe identifié par son numéro (N_§) est compris dans les paragraphes qui appartiennent à la réponse.

Postcondition :

- Si N_§ est attribué et existe, Exist = true et Fin_f = false
sinon Exist = false et Fin_f = false.
Si le numéro n'est pas attribué Fin_f = true.

```
* Exist_Hier ( N_Test, N_Pers : integer; Hier : string[Lid];  
               var Exist, Fin_f : boolean ; Typ_R : char);
```

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Détermine si Hier est une hiérarchie déjà attribuée à un élément de la réponse (c-à-d : un paragraphe) correspondant aux numéros de test et de personnes donnés (N_Test et N_Pers).

Postcondition :

- Si Hier est attribué et existe, Exist = true et Fin_f = false.
sinon Exist = false et si la hiérarchie est attribuée, Fin_f = false sinon, fin_f = true.

```
* Hierar ( N_Test, N_Pers, Hier : integer; var N_§ : integer;  
           var Id-§ : string[Lid]; Typ_R : char);
```

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.
- Hier est une hiérarchie attribuée à un paragraphe de la réponse.

Fonction :

Fournir le numéro de paragraphe (N_§) et l'identifiant de celui-ci (Id_§).

Postconditions :

- N_§ contient le numéro du paragraphe correspondant et Id_§ contient son identifiant.

```
* Aj_Hier ( N_Test, N_Text, N_Pers, N_§ : integer ; Id_§, Hier :  
            string[Lid]; Typ_R : char);
```

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- La hiérarchie Hier est une hiérarchie identifiante non encore attribuée à un paragraphe du texte de travail.
- Le numéro N_§ est un numéro identifiant de paragraphe non encore attribué à un paragraphe du texte de travail.
- Le numéro de texte, s'il est <> 0, est un numéro de texte associé à la hiérarchie introduite.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Ajoute un élément au fichier de réponse correspondant aux numéros identifiants du test et de la personne.

L'élément est formé d'un identifiant de paragraphe (Id_§), d'un numéro de paragraphe (N_§) et d'une hiérarchie (Hier) et d'un texte explicatif.

Postcondition :

- L'élément a bien été ajouté.

```
* Del_Hier ( N_Test, N_Pers, N_§ : integer; Hier : string[Lid];  
            Typ_R : char);
```

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.

- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- La variable Hier est une hiérarchie identifiant un paragraphe du texte de travail.
- Soit N_§ <> 0 et Hier = ' ' ,
soit N_§ = 0 et Hier <> ' ' .

Fonction :

Eliminer de la réponse l'enregistrement correspondant au numéro de paragraphe donné (N_§) ou à une hiérarchie de paragraphe donné (Hier).

Postconditions :

L'élément correspondant à un numéro de paragraphe (N_§) ou à une hiérarchie de paragraphe (Hier) n'appartient plus à la réponse traitée.

4.5 COLLECTION CONTENANT LE DERNIER NUM. ATTRIBUÉ à UN TEST ET LE DERNIER NUM. ATTRIBUÉ à UNE PERSONNE

* Lir_N_Test (var N_Test : integer);

Précondition : /

Fonction :

Lire le dernier numéro attribué à un test.

Postcondition :

- N_Test contient le numéro attribué au dernier test introduit.

* Lir_N_Pers (var N_Pers : integer);

Précondition : /

Fonction :

Lire le dernier numéro attribué à une personne testée.

Postcondition :

- N_Pers contient le numéro attribué à la dernière personne testée introduite.

* Lir_N_Ped (var N_Ped : integer);

Précondition : /

Fonction :

Lire le dernier numéro attribué à une personne pédagogue.

Postcondition :

- N_Ped contient le numéro attribué à la dernière personne introduite.

* Chge_N_Test (N_Test : integer);

Précondition :

- N_Test est le dernier numéro attribué à un test.

Fonction :

Inscrit dans le fichier F_Num le dernier numéro attribué à un test.

Postcondition :

- F_Num contient le numéro attribué au dernier test introduit.

* Chge_N_Pers (N_Pers : integer);

Précondition :

- N_Pers est le dernier numéro attribué à une personne testée.

Fonction :

Inscrit, dans le fichier F_Num, le dernier numéro attribué à une personne testée.

Postcondition :

- F_Num contient le numéro attribué à la dernière personne testée introduite.

* Chge_N_Ped (N_Ped : integer);

Précondition :

- N_Ped est le dernier numéro attribué à un pédagogue.

Fonction :

Inscrit, dans le fichier F_Num, le dernier numéro attribué à une personne pédagogue.

Postcondition :

- F_Num contient le numéro attribué à la dernière personne pédagogue introduite.

4.6 COLLECTIONS CHRONOLOGIQUES

Pour les réponses de type hiérarchie :

- - - - -

```
* Lir_E_H ( var N_§ : integer; var Hier : string[Lid]; var N_Choix
           : integer; N_Plac, N_Test, N_Pers : integer; Typ_R:char);
```

Préconditions :

- N_Test est un numéro attribué à un test existant,
- N_Pers est un numéro attribué à une personne existante,
- N_Plac entier ≥ 0 ,
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Lire l'élément du fichier chronologique correspondant aux identifiants internes de test et de personne donnés (N_Test et N_Pers) et qui se trouve à la place N_Plac.

Postcondition :

Les paramètres N_§, Hier et N_Choix contiennent les parties formant l'élément du fichier à la place N_Plac.

Si N_Plac dépasse les limites du fichier, N_§ = 0.

```
* Ecrire_E_H ( N_§: integer; Hier : string[Lid]; N_Choix : integer
              N_test, N_Pers : integer; Typ_R : char);
```

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.

- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Ecrire à la fin du fichier chronologique correspondant aux identifiants interne de test et de personne donnés (N_Test et N_Pers) un nouvel élément. Cet élément est formé d'un numéro interne de paragraphe N_§, d'un numéro de hiérarchie Hier et d'un numéro de choix N_Choix.

Postcondition :

Le fichier chronologique concerné possède un élément en plus qui contient un record ayant les valeurs des paramètres N_§, Hier et N_Choix.

* Del_Chr_H (N_Test, N_Pers : integer; Typ_R : char);

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Détruire le fichier choronologique désigné par N_Test et N_Pers et Typ_R.

Postcondition :

Le fichier a bien été détruit.

Pour les réponses de type liaison :

* Lir_E_L (var N_Ele1, N_Ele2, N_Rel, N_Choix : integer;
N_Plac, N_Test, N_Pers : integer; Typ_R : char);

Préconditions :

- N_Test est un numéro de test existant
- N_Pers est un numéro de personne existante
- N_Plac est un entier ≥ 0

- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Lire l'élément du fichier chronologique correspondant aux identifiants internes de test et de personne donnés (N_Test et N_Pers) et qui se trouve à la place N_Plac.

Postconditions :

Les paramètres N_Ele, T_Ele et N_Choix contiennent les parties formant l'élément du fichier à la place N_Plac.

* Ecrire_E_L (N_Ele1, N_Ele2, N_Rel, N_Choix, N_Test, N_Pers : integer; Typ_R : char);

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.
- Si N_Ele1 est un numéro de paragraphe (≥ 0) ou d'une occurrence de rectangle introduit dans la réponse (< 0), Ele2 = N_Rel = 0;
- Si N_Ele1 est un numéro de paragraphe (≥ 0) ou d'occurrence de rectangle origine (< 0) d'une relation cible, N_Ele2 = num. paragraphe cible et N_Rel = num. de la relation employée.

Fonction :

Ecrire à la fin du fichier chronologique correspondant aux identifiants internes de test et de personne donnés (N_Test et N_Pers) un nouvel élément. Cet élément est formé d'un numéro interne d'élément N_Ele, d'un indicateur de type de l'élément T_Ele et d'un type de choix N_Choix;

Postconditions :

Le fichier chronologique concerné possède un élément en plus qui contient un record ayant les valeurs des paramètres N_Ele, T_Ele et Choix.

* Del_E_L (N_Test, N_Pers : integer; Typ_R : char);

Préconditions :

- N_Test est un numéro attribué à un test existant.
- N_Pers est un numéro attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Détruire le fichier chorologique désigné par N_Test et N_Pers.

Postcondition :

Le fichier a bien été détruit.

Pour les réponses de type réseaux :

- - - - -

* Lir_E_R (var N_Ele, N_O, N_C : integer; N_Plac, N_Test, N_Pers : integer; var N_Choix : integer ; Typ_R : char);

Préconditions :

- N_Test est un numéro attribué à un test existant.
- N_Pers est un numéro attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Lire l'élément du fichier chronologique correspondant aux identifiants internes de test et de personne donnés (N_Test et N_Pers) et qui se trouve à la place N_Plac.

Postconditions :

Les paramètres N_Ele1, N_Ele2, N_Rel et N_Choix contiennent les parties formant l'élément du fichier à la place N_Plac.

Si N_Plac dépasse les limites du fichier, N_Ele1 = 0.

* Ecrire_E_R (N_Ele1, N_Ele2, N_Rel, N_Choix, N_Test, N_Pers :
integer; Typ_R : char);

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Ecrire à la fin du fichier chronologique correspondant aux identifiants internes de test et de personne donnés (N_Test et N_Pers) un nouvel élément. Cet élément est formé des éléments N_Ele1, N_Ele2, N_Rel et N_Choix;

Postconditions :

Le fichier chronologique concerné possède un élément en plus qui contient un record ayant les valeurs des paramètres N_Ele1, N_Ele2, N_Rel et N_Choix.

* Del_E_R (N_Test, N_Pers : integer; Typ_R : integer);

Préconditions :

- Le numéro N_Test est un numéro identifiant attribué à un test existant.
- Le numéro N_Pers est un numéro identifiant attribué à une personne existante.
- Typ_R contient soit P soit T soit E suivant que la réponse est du type pédagogue, testé ou exemple.

Fonction :

Détruire le fichier choronologique désigné par N_Test et N_Pers.

Postcondition :

Le fichier a bien été détruit.

4.7 COLLECTION DES PERSONNES TESTÉES

* Aj_Pers (Id_Pers : string[Lid] ; N_Pers : integer);

Préconditions :

- Id_Pers ne correspond à aucune personne existante.
- N_Pers est le numéro à attribuer au nouvel enregistrement.

Fonction :

Ajoute Id_Pers à la fin du fichier des personnes existantes et attribue un numéro identifiant à Id_Pers.

Postcondition :

- Id_Pers fait partie du fichier des personnes existantes et N_Pers est le numéro identifiant qui lui est attribué.

* Exist_Pers (Id_Pers :string[Lid] ; var N_Pers :integer ;
var Exist, Fin_f : boolean);

Préconditions :

- Id_Pers <> ' ' et N_Pers < 0
- ou Id_Pers = ' ' et N_Pers >= 0

Fonction :

Détermine si l'utilisateur dont on a donné soit l'intitulé Id_Pers soit le numéro N_Pers existe dans le fichier des personnes testées.

Si c'est intitulé qui est fourni et que la personne existe alors, son numéro est fourni par la procédure.

Postcondition :

- Si la personne existe, Exist = true, Fin_f = false et N_Pers contient son numéro.
- sinon, Exist = false, N_Pers est indéterminé et si l'élément n'appartient pas au fichier Fin_f = true si non Fin_f = false.

* Del_Pers (Id_Pers : string[Lid] ; N_Pers : integer);

Préconditions :

- Id_Pers <> ' ' et identifie une personne existante et N_Pers < 0
- ou N_Pers >= 0 et identifie une personne existante et Id_Pers = ' ' ,

Fonction :

Elimine la personne dont on a donné le numéro N_Pers ou l'intitulé Id_pers du fichier des personnes testées.

Postcondition :

- La personne déterminée n'appartient plus au fichier des personnes testées.

* List_TPers (var Tab_T_P : array [1..NbTest] of integer ;
N_Pers : integer);

Précondition :

- N_Pers est un numéro interne identifiant une personne existante.

Fonction :

Fournir la liste des tests existants réalisés par la personne qui est identifiée par N_Pers.

Postcondition :

- la liste des tests réalisés par la personne déterminée a été fournie et est contenue dans Tab_T_P.

* Del_TPers (N_Test, N_Pers : integer);

Préconditions :

- N_Pers est le numéro interne identifiant d'une personne existante.
- N_Test est le numéro interne identifiant d'un test existant.

Fonction :

Détruire, dans la liste des tests réalisés par la personne dont le numéro est N_Pers, l'indication, si elle existe, que le test identifié par N_Test a été réalisé.

Postcondition :

- le test dont le numéro est N_Test ne fait pas partie de la liste des tests réalisés par la personne dont le numéro est N_Pers.

* Aj_TPers (N_Test, N_Pers : integer);

Préconditions :

- N_Pers est un numéro identifiant une personne existante.
- N_Test est un numéro identifiant un test existant.
- La personne identifiée par N_Pers n'a pas dans sa liste des tests réalisés le test dont l'identifiant est N_Test.

Fonction :

Ajoute à la liste des tests existants réalisés par la personne identifiée par N_Pers, le numéro identifiant du test N_Test.

Postcondition :

- N_Test se trouve dans la liste des tests réalisés par la personne identifiée par N_Pers.

* Exist_TPers (N_Test, N_Pers : integer; var Exist : boolean);

Préconditions :

- N_Test est un numéro interne identifiant un test existant.
- N_Pers est un numéro interne identifiant une personne existante.

Fonction :

Détermine si le test dont le numéro identifiant est N_Test fait partie des tests réalisés par la personne dont le numéro identifiant est N_Pers.

Postcondition :

- Si la personne déterminée par N_Pers a réalisé le test déterminé par N_Test, Exist = true, si non, Exist = false.

* Lir_Pers (N_Pers integer; var Id_Pers : string);

Précondition :

- N_Pers est un numéro attribué à une personne existante.

Fonction :

Fournir l'identifiant et la notion d'existence d'une personne dont le numéro est N_Pers.

Postcondition :

- Id_Pers contient l'identifiant de la personne dont le num. est N_Pers.

* Nb_TPers (N_Pers : integer; var Nb_Test : integer);

Précondition :

- N_Pers est un numéro identifiant une personne testée existante.

Fonction :

Donner le nombre de tests réalisés et conservés par une personne testée dont le numéro identifiant est N_Pers.

Postcondition :

- Nb_Test contient le nombre de tests réalisés et conservés de la personne testée dont le numéro est N_Pers.

4.8 COLLECTION DES PÉDAGOGUES

* Aj_Ped (Id_Ped : string[Lid] ; N_Ped : integer);

Précondition :

- Id_Ped ne correspond à aucune personne existante.
- N_Ped est le numéro à attribuer au nouvel enregistrement.

Fonction :

Ajoute Id_Ped à la fin du fichier des personnes existantes et attribue un numéro identifiant à Id_Ped.

Postcondition :

- Id_Ped fait partie du fichier des personnes existantes et N_Ped est le numéro identifiant qui lui est attribué

```
* Exist_Ped (Id_Ped :string[Lid] ; var N_Ped : integer ;  
              var Exist, Fin_f : boolean );
```

Préconditions :

- Id_Ped <> ' ' et N_Ped < 0
- Id_Ped = ' ' et N_Ped >= 0

Fonction :

Détermine si l'utilisateur dont on a donné soit l'intitulé Id_Ped soit le numéro N_Ped existe dans le fichier des pédagogues.

Si c'est l'intitulé qui est fourni et que le pédagogue existe, la procédure dourni son numéro N_Ped.

Postcondition :

- Si le pédagogue existe, Exist = true et N_Ped contient son numéro,
sinon, N_Ped est indéterminé et si l'élément n'est pas contenu dans le fichier, Fin_f = true et Exist = false, sinon Fin_f = false et Exist = false.

```
* Del_Ped ( Id_Ped : string[Lid] ; N_Ped :integer );
```

Préconditions :

- Id_Ped <> ' ' et identifie un pedagogue existant et N_Ped < 0
- N_Ped >= 0 et identifie un pédagogue existant et Id_Ped = ' ' ,

Fonction :

Elimine le pédagogue dont on a donné le numéro N_Ped ou l'intitulé Id_ped du fichier des pédagogues.

Postcondition :

- La personne déterminée n'appartient plus au fichier des pédagogues.

* Lir_Ped (N_Ped : integer; var Id_Ped : string);

Précondition :

- N_Ped est un numéro attribué à un pédagogue existant.

Fonction :

Fournir l'identifiant d'un pédagogue dont le numéro est N_Ped.

Postcondition :

- Id_Ped contient l'identifiant du pédagogue dont le num. est N_Ped.

4.9 COLLECTIONS DES TEXTES

* Aj_Titr (N_Test, T_Txt, N_Pers, N_Txt : integer; F_Titr :
file of char ; Entet : boolean);

Préconditions :

- N_test est un numéro identifiant un test existant
- T_Txt est soit = 1 si le texte est de type travail
= 2 explicatif
= 3 réponse
- N_Txt est nul si T_Txt est égal à 1 ou 2 et est différent de 0 si T_Txt égal 3.
- Entet = true si le titre est celui de l'entête, false si non.
- N_Pers contient le numéro identifiant une personne existante si T_Txt = 3 et est indéterminé si T_Txt = 1.
- F_Titr est le nom d'un fichier de texte non vide.

Fonction :

Ajoute à la fin du fichier texte désigné par N_Test, N_Pers, T_Txt, N_Txt, le titre de paragraphe contenu dans le fichier qui a pour nom F_Titr.

Postconditions :

Le titre de paragraphe contenu dans le fichier F_Titr se trouve en fin du fichier de texte concerné.


```
* Aj_Par_tet ( N_Test, N_Pers, T_Txt, N_Txt : integer; F_s :
               file of char, Entet : boolean);
```

Préconditions :

- N_test est un numéro identifiant un test existant
- T_Txt est soit = 1 si le texte est de type travail
 = 2 explicatif
 = 3 réponse
- N_Txt est nul si T_Txt est égal à 1 ou 2 et est différent de 0 si T_Txt égal 3.
- N_Pers contient le numéro identifiant une personne existante si T_Txt = 3 et est indéterminé si T_Txt = 1 ou 2.
- F_Titr contient le nom d'un fichier de texte non vide.
- Entet = true si le titre est celui de l'en tête, false si non.

Fonction :

Ajoute à la fin du fichier texte désigné par N_Test, N_Pers, T_Txt, N_Txt, le paragraphe contenu dans F_Titr.

Postconditions :

Le paragraphe contenu dans le fichier F_Titr se trouve en fin du fichier de texte concerné.

```
* Del_Titr (N_Test, T_Txt, N_Pers, N_Txt : integer; F_Titr :
            file of char ; Entet : boolean);
```

Préconditions :

- N_test est un numéro identifiant un test existant
- T_Txt est soit = 1 si le texte est de type travail
 = 2 explicatif
 = 3 réponse
- N_Txt est nul si T_Txt est égal à 1 ou 2 et est différent de 0 si T_Txt égal 3.
- N_Pers contient le numéro identifiant une personne existante si T_Txt = 3 et est indéterminé si T_Txt = 1 ou 2.
- F_s contient le nom d'un fichier de texte non vide.
- Entet = true si le titre est celui de l'en tête, false sinon.

Fonction :

Détruit le titre de paragraphe donné dans le texte déterminé par les valeurs de N_Txt, N_Pers, T_Txt, T_Test.

Postconditions :

Le titre donné a bien été détruit.

```
* Del_Par_Tet ( N_Test, N_Pers, T_Txt, N_Txt integer; F_s :  
                file of char ; Entet : boolean);
```

Préconditions :

- N_test est un numéro identifiant un test existant.
- T_Txt est soit = 1 si le texte est de type travail
 = 2 explicatif
 = 3 réponse
- N_Txt est nul si T_Txt est égal à 1 ou 2 et est différent de 0 si T_Txt égal 3.
- N_Pers contient le numéro identifiant une personne existante si T_Txt = 3 et est indéterminé si T_Txt = 1 ou 2.
- F_s contient un nom de fichier de texte non vide.
- Entet = true si le titre est celui de l'entête, false sinon.

Fonction :

Détruit le paragraphe donné dans le texte déterminé par les valeurs de N_Txt, N_Pers, T_Txt, T_Test.

Postconditions :

Le paragraphe donné a bien été détruit.

```
* Lir_Titr_Tet (N_Test, T_Txt, N_Pers, N_Txt, N_s : integer;  
                Entet : boolean; F_Titr : string);
```

Préconditions :

- N_test est un numéro identifiant un test existant
- T_Txt est = 1 : le texte est de type travail
- N_Txt est nul.
- N_Pers contient une valeur indéterminée.
- F_Titr contient un nom de fichier de texte non vide.
- Entet = true si le titre est celui de l'entête, false si non.

Fonction :

Lire le titre indiqué par la valeur de Entet et celle de N_Par.

Postcondition :

- Si le texte indiqué n'existe pas, le fichier F_Titr est vide, sinon ce fichier contient le titre lu.

4.10 COLLECTION DES RÉPONSES DE TYPE RÉSEAU

* Aj_CO (N_Test, N_Pers, N_Txt, N_C_O : integer; Typ_R :boolean);

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- N_Txt est une variable qui peut contenir un numéro de texte associé au concept cible introduit. Si aucun texte n'est associé, N_Txt = 0.
- N_C_O est un numéro interne identifiant un concept utilisable pour le test de numéro N_Test.
- Entet = true si le titre est celui de l'en tête, false sinon.

Fonction :

Ajoute un élément supplémentaire à la réponse traitée. Cet élément est un concept origine dont le numéro interne identifiant est N_C_O.

Si l'introduction de cet élément est accompagnée d'un texte, celui-ci est également enregistré.

Postconditions :

- N_C_O fait partie de la réponse traitée et si un texte a été enregistré, celui-ci fait également partie de la réponse.

* Aj_LRec (N_Test, N_Pers, N_Txt, N_Occ, N_L_Rec: integer ; Typ_R : string ; F_CRec : string);

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.

- N_Pers est un numéro interne identifiant une personne existante.
- N_Txt est une variable qui peut contenir un numéro de texte associé à la liaison rectangle introduite. Si aucun texte n'est associé, N_Txt = 0.
- N_L_Rec est un numéro interne identifiant une liaison de type rectangle utilisable pour le test de numéro N_Test.
- Entet = true si le titre est celui de l'en tête, false sinon.
- F_CRec contient le nom d'un fichier texte non vide.
- N_Occ est le numéro d'occurrence de la liaison rectangle.

Fonction :

Ajoute un élément supplémentaire à la réponse traitée. Cet élément est une occurrence de la liaison rectangle dont le numéro interne identifiant est N_L_Rec et le numéro d'occurrence est N_Occ et qui contient les concepts qui se trouvent dans le fichier dont le nom est F_CRec.

Si l'introduction de cet élément est accompagnée d'un texte, celui-ci est également enregistré.

Postconditions :

- N_Occ et N_L_Rec font partie de la réponse traitée et si un texte a été enregistré, celui-ci fait également partie de la réponse. Les numéros de concepts appartenant à l'occurrence de la liaison rectangle ont été enregistrés dans la réponse.

* Aj_CFle (N_Test, N_Pers, N_Txt, N_Rel, N_C, N_O : integer;
Typ_R : boolean);

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- N_Rel est un numéro interne identifiant une liaison de type flèche utilisable pour le test de numéro N_Test.
- N_Txt est une variable qui peut contenir un numéro de texte associé à la liaison introduite. Si aucun texte n'est associé, N_Txt = 0.
- N_C est un numéro interne identifiant un concept ou un rectangle cible existant pour le test réalisé.
(si c'est un rectangle, cible < 0).
- Typ_R contient le caractère P si la réponse est du type testé, E si la réponse est du type exemple, T si la

réponse est du type réponse-type.

- N_O est un numéro interne identifiant un concept ou une occurrence de liaison rectangle cible.(si occ. rect. <0)

Fonction :

Ajoute un élément supplémentaire à la réponse traitée. Cet élément est une liaison flèche dont le numéro interne identifiant est N_Rel partant d'un élément origine dont le num est N_O et arrivant à un élément cible N_C.

Si l'introduction de cet élément est accompagnée d'un texte, celui-ci est également enregistré.

Postconditions :

- N_Rel fait partie de la réponse traitée et si un texte a été enregistré, celui-ci fait également partie de la réponse ainsi que le concept ou l'occurrence de liaison rectangle cible N_C.

* Del_Con (N_Test, N_Pers, N_C : integer ; Typ_R : boolean);

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- N_C est un numéro interne identifiant un concept utilisé par la personne identifiée par N_Pers pour le test de numéro N_Test.
- Typ_R contient le caractère P si la réponse est du type testé, E si la réponse est du type exemple, T si la réponse est du type réponse-type.

Fonction :

Détruire le concept N_C utilisé, dans la réponse traitée, comme origine ou comme cible. La destruction du concept peut entraîner la destruction d'autres éléments tels que liaisons ou textes.

Postconditions :

Le concept (N_C) à détruire n'appartient plus à la réponse traitée. Les éléments liés à ce concept ont été supprimés également.

* Del_Rec (N_Test, N_Pers, N_Rec : integer; Typ_R : boolean);

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- N_Rec est un numéro interne identifiant une occurrence de la liaison rectangle utilisée par la personne identifiée par N_Test pour le test de numéro N_Test.
- Typ_R contient le caractère P si la réponse est du type testé, E si la réponse est du type exemple, T si la réponse est du type réponse-type.

Fonction :

Détruire l'occurrence de la liaison rectangle désignée par N_Rec dans la réponse traitée.

Postcondition :

La liaison rectangle désignée par N_Rec n'existe plus dans la réponse traitée.

* Del_Txt (N_Test, N_Pers, N_Txt : integer ; Typ_R : boolean);

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- N_Txt est une variable contenant un numéro de texte existant pour la réponse identifiée par N_Test et N_Pers.
- Typ_R contient le caractère P si la réponse est du type testé, E si la réponse est du type exemple, T si la réponse est du type réponse-type.

Fonction :

Détruire la référence numérique du test donné dans la réponse traitée.

Postcondition :

La référence au texte dont le numéro est N_Txt a bien été détruite dans la réponse traitée.

* Lir_CO (N_Test, N_Pers, F_Ele_O : string ; Typ_R : boolean);

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- F_Ele_O est un fichier contenant des records formés d'un numéro de concept origine et d'un numéro de texte associé à ce concept s'il existe. (s'il n'existe pas cette partie est nulle). F_Ele_O file of Ele_O.

Fonction :

Lire dans le fichier réponse correspondant aux valeurs de N_Test et N_Pers, l'ensemble des numéros identifiants les concepts-origine de cette réponse.

Postconditions :

- F_Ele_O est un fichier contenant des records formés d'un numéro de concept origine et d'un numéro de texte associé à ce concept s'il existe. (s'il n'existe pas cette partie est nulle). F_Ele_O file of Ele_O.

* Lir_Cib (N_Test, N_Pers, N_C :integer; F_Ele_C : file of Ele_C;
Con : boolean; Typ_R : Char);

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- N_C est un numéro interne identifiant un concept utilisé dans la réponse identifiée par N_Pers et N_Test comme concept origine ou identifie une occurrence de la liaison rectangle employée comme "concept origine".
- Typ_R contient le caractère P si la réponse est du type testé, E si la réponse est du type exemple, T si la réponse est du type réponse-type.
- Con est true si l'élément origine est un concept, false si c'est une occurrence de la liaison rectangle.

Fonction :

Lire dans le fichier réponse correspondant aux valeurs de N_Test et N_Pers :

- l'ensemble des numéros identifiants les éléments cibles correspondant au concept origine dont le numéro est N_C.
- le numéro de la relation flèche utilisée.
- le texte associé à la liaison, s'il existe.

Postcondition :

- F_Ele_C est un fichier contenant la liste des éléments cibles d'un élément origine donné N_O et du numéro de liaison flèche utilisée pour chaque concept cible ainsi que des textes, s'il existent, qui peuvent être associés aux liaisons employées.

```
* Lir_Con_Rec (N_Test, N_Pers, N_Occ : integer; var N_Txt :  
               integer; F_C_Rec : string; Typ_R : boolean);
```

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- N_Occ est un numéro d'occurrence de liaison rectangle contenu dans la réponse identifiée par N_Test et N_Pers.
- Typ_R contient le caractère P si la réponse est du type testé, E si la réponse est du type exemple, T si la réponse est du type réponse-type.

Fonction :

Lire dans le fichier réponse, correspondant aux valeurs de N_Test et N_Pers, l'ensemble des concepts appartenants à l'occurrence de la liaison rectangle dont le numéro est N_Occ.

Lire le texte associé à l'occurrence de la liaison rectangle dont le numéro est N_Occ, s'il existe.

Postcondition :

- F_CRec est le nom du fichier qui contient les données sur les concepts contenus dans l'occurrence N_Occ de la liaison rectangle.


```
* Aj_LRec_O ( N_Test, N_Pers, N_Txt, N_Occ, N_Rec, N_C, N_Rel :  
integer; Typ_R char);
```

Préconditions :

- N_Test est un numéro identifiant interne correspondant à un test existant.
- N_Pers est un numéro interne identifiant une personne existante.
- N_Occ est un numéro d'occurrence de liaison rectangle contenu dans la réponse identifiée par N_Test et N_Pers.
- Typ_R contient le caractère P si la réponse est du type testé, E si la réponse est du type exemple, T si la réponse est du type réponse-type.
- N_Txt est un numéro de texte associé à l'introduction de la liaison rectangle.(si pas de texte N_Txt = 0)
- N_C est le numéro de l'élément cible (< 0 si occurrence de liaison rectangle si non >= 0).
- N_Rel est le numéro de la liaison flèche utilisée.
- N_L_Rel numéro attribué à la liaison rectangle pour le test dont le numéro est N_Test.

Fonction :

Insérer dans la réponse une nouvelle information qui est que l'occurrence de la liaison rectangle dont le numéro est N_Occ est l'origine d'une relation flèche vers un élément dont le numéro est N_C (<0 si occurrence de la liaison rect.). La relation flèche utilisée porte le numéro N_Rel. Un texte explicatif peut être associé à cette introduction.

Postcondition :

- l'élément a bien été enregistré.

Annexe 5 : Presentation de l'implémentation réalisée

Le lecteur qui voudra prendre connaissance du code des parties qui ont été réalisées pourra s'adresser à Monsieur Cherton.

Le code n'a pas été placé dans cette annexe pour une question de volume. De plus, le mémorant désire continuer la codification de certaines parties.

Annexe 6 : Manuel d'utilisation

INTRODUCTION

Ce manuel est divisé en deux parties.

La première présente en détail la marche-à-suivre pour :

- mettre en route le logiciel
- introduire un nouveau test
- réaliser un test
- analyser des résultats ou détruire des réponses, des tests

Cette première partie est pour l'instant incomplète. Car les deux derniers points : réalisation et analyse de tests n'ont pas été implémentés.

La seconde partie reprend toutes les manipulations offertes aux utilisateurs sur les différents tests, textes et réponses contenus dans le logiciel.

Cette seconde partie a été réalisée pour permettre à un utilisateur fréquent du logiciel de retrouver les manipulations possibles sans lire la marche-à-suivre détaillée.

PREMIÈRE PARTIE : MARCHE-à-SUIVRE

1) Mise en route du logiciel

Pour mettre en route le logiciel, il faut :

- allumer l'ordinateur
- si le macintosh utilisé n'est pas du type 400K :
 - * introduire la disquette 'syst. 400K' dans le drive principal. Lorsque le système est chargé, éjecter la disquette.
- introduire la disquette 'pascal' dans la drive principal.
- introduire la disquette 'mémo' dans le second drive.
- un message est affiché demandant de choisir entre :
 - * introduire un nouveau test,
 - * réaliser un test existant,
 - * Analyser des réponses ou détruire des éléments,
 - * arrêter.

Après ces opérations, le logiciel est en état d'exécution.

2) Introduction d'un test dans le logiciel

L'action 'introduction d'un nouveau test' doit avoir été choisie au début de l'exécution du logiciel.

Si c'est le cas, un message est affiché demandant l'introduction d'un nom de pédagogue existant (6 caractères maximum) pour que vous ayez le droit d'accéder à cette partie du logiciel.

- Si le nom introduit correspond à un nom de pédagogue contenu dans le logiciel, l'accès est accordé.

Si non, un message d'erreur signale que le nom donné n'existe pas et que vous avez droit à un second essai. Si le second essai est mauvais, vous êtes éjecté.

Cliquez la zone OK de l'écran d'erreur pour faire disparaître l'écran d'erreur et procédez à la seconde introduction du nom de pédagogue.

Les explications suivantes sont valables dans le cas où l'accès est accordé.

Un message est affiché demandant si vous désirez consulter la liste des noms des tests existants dans le logiciel.

- si oui, cliquez la zone 'oui' sur l'écran.

La liste des écrans est alors présentée. Vous avez à votre disposition les manipulations possibles, en consultation, sur une liste d'éléments. (voir la liste de toutes les manipulations possibles à la fin du manuel)

- si non, cliquez 'non' sur l'écran.

Un message demandant l'introduction du nom identifiant le nouveau test est affiché. (ce nom a 6 caractères au maximum)

- si le nom introduit est déjà attribué, c-à-d : identifie un test existant, un message est affiché qui le signale et qui vous demande de réintroduire un nom de test ou d'annuler la commande en tapant directement 'RETURN'.
- s'il n'y a pas de nom de test introduit, la demande d'introduction est annulée et vous êtes replacé au niveau du premier menu.
- si le nom est accepté, vous passez à l'introduction du type du test.

Si le nom est accepté, un message est affiché demandant l'introduction du type du test dont l'identifiant a été introduit.

Ce type est soit 'réseau' -> introduire 'r'
'liaison' -> introduire 'l'
'hiérarchie' -> introduire 'h'

- si vous introduisez plusieurs caractères seul le premier est enregistré.
- si le caractère introduit est différent de 'r' ou 'l' ou 'h', un message est affiché signalant que le type introduit est incorrect et demande la réintroduction du type. Ceci sera effectué tant que le type est incorrect.

Lorsque les noms et type du test ont été introduits et acceptés, vous allez procéder à l'introduction des différentes parties du test. Ces parties sont :

- un texte explicatif, s'il existe

- le texte de travail
- les matériaux du test, s'il est de type 'réseau' ou 'liaison'.
- un exemple de réponse, s'il existe
- une réponse-type

2.1) Introduction du texte explicatif

- - - - -

Un message est affiché demandant si vous voulez introduire un texte explicatif.

- si non, cliquez 'NON' et vous passez à l'introduction du texte de travail.
- si oui, cliquez 'OUI'.

Un texte explicatif est formé d'une suite non vide de paragraphes qui n'ont pas de titre.

Tant que vous ne signalez pas la fin de l'introduction du texte, un message va apparaître demandant l'introduction d'un paragraphe. L'introduction du paragraphe s'effectue avec un passage à la ligne automatique. Vous ne tapez sur la touche RETURN que pour signaler la fin du paragraphe. La fin de l'introduction du texte est signalée par l'introduction d'un paragraphe vide. (c-à-d : tapez 'RETURN' sans introduire aucun caractère auparavant)

Vous avez la possibilité de signaler que un mot ou une suite de mots ou de caractères doivent être présentés soulignés en tapant le caractère ' avant et après l'élément à souligner.

2.2) Introduction du texte de travail

- - - - -

Un message est affiché signalant que vous allez introduire votre texte de travail. Cliquez 'suite' pour passer à son introduction.

Un texte de travail doit être formé d'un paragraphe au moins. Chaque paragraphe est formé d'un titre (facultatif) et d'un corps (obligatoire).

Lors de l'introduction, les deux messages d'introduction suivants vont se répéter jusqu'au moment où vous taperez le signal de fin d'introduction. Ce signal consiste en l'introduction d'un paragraphe sans titre et sans corps.

- le premier message affiché demande l'introduction du titre d'un paragraphe (maximum 20 caractères).
- * si le paragraphe n'a pas de titre, tapez 'RETURN' directement.
- * si le paragraphe a un titre introduisez le puis tapez

'RETURN'.

si le titre a plus de 20 caractères un message d'erreur est affiché demandant la réintroduction du titre.

si le titre a déjà été attribué à un autre paragraphe introduit pour le texte de travail traité, un message d'erreur est affiché demandant la réintroduction du titre.

- le second message affiché demande l'introduction du corps du paragraphe se rapportant au titre qui vient d'être introduit, s'il existe.

L'introduction se déroule avec un passage automatique à la ligne. Tapez 'RETURN' pour signaler la fin du paragraphe.

si le corps du paragraphe introduit est vide et qu'un titre a été introduit pour ce paragraphe, un message d'erreur est affiché demandant l'introduction du corps du paragraphe.

si le corps du paragraphe introduit et le titre qui s'y rapporte sont vides, l'introduction du texte est terminée.

si le corps du paragraphe est non vide, celui-ci est enregistré.

2.3) Introduction des matériaux de test

A) Dans le cas de test de type 'réseau de concepts' :

.....
Un message est affiché qui signale que vous allez introduire les matériaux du test. Cliquez 'suite' pour commencer.

- introduction des concepts :

Les premiers matériaux introduits sont les concepts. Un concept au moins doit être introduit et accepté. Un concept est formé de deux suites de caractères qui l'identifient chacune.

Deux messages vont se succéder tant que vous n'aurez pas donné le signal de fin.

* le premier message demande l'introduction de la suite de caractères identifiante contenant la signification en langue naturelle d'un concept (20 caractères maximum).

si la suite introduite est non identifiante par rapport aux concepts déjà introduits pour ce test, un message d'erreur est affiché signalant que la suite est refusée car non identifiante et demande une réintroduction.

- # si la suite introduite fait plus de 20 caractères, un message d'erreur est affiché signalant le refus de la suite et demande une réintroduction.
- # si la suite introduite est vide, l'introduction des concepts est terminée.
- # si non, la suite est acceptée.

* le second message demande l'introduction de la suite identifiante contenant un rappel mnémotechnique de la signification du concept (6 caractères maximum);

- # si la suite introduite est non identifiante par rapport aux concepts déjà introduits pour ce test, un message d'erreur est affiché signalant que la suite est refusée car non identifiante et demande une réintroduction.
- # si la suite introduite fait plus de 6 caractères, un message d'erreur est affiché signalant le refus de la suite et demande une réintroduction.
- # si la suite introduite est vide et que la première est non vide, un message d'erreur est affiché demandant d'introduire une suite identifiante non vide.
- # si non, la suite est acceptée.

- introduction des liaisons :

Les seconds matériaux introduits sont les liaisons. Une liaison au moins doit être introduite et acceptée.

Un message est affiché demandant, si pour le test introduit, la liaison 'rectangle' est utilisée ou non.

- * si oui, cliquez 'OUI' sur l'écran,
- si non, cliquez 'NON' sur l'écran.

Introduction des liaisons flèches :

Une liaison flèche est formée de deux suites de caractères qui l'identifient chacune.

Deux messages vont se succéder tant que vous n'aurez pas donné le signal de fin.

- * le premier message demande l'introduction de la suite identifiante de caractères qui contient la signification en langue naturelle d'une liaison (20 caractères maximum).
- # si la suite introduite est non identifiante par rapport aux liaisons et concepts déjà introduits pour ce test, un message d'erreur est affiché signalant que la suite est refusée car non identifiante et demande une réintroduction.
- # si la suite introduite fait plus de 20 caractères, un message d'erreur est affiché signalant le refus de la

- suite et demande une réintroduction.
- # si la suite introduite est vide, l'introduction des liaisons est terminée.
 - # sinon, la suite est acceptée.
- * le second message demande l'introduction de la suite identifiante de caractères qui contient un rappel mnémotechnique de la signification de la liaison (6 caractères maximum).
- # si la suite introduite est non identifiante par rapport aux concepts et aux liaisons déjà introduits pour ce test, un message d'erreur est affiché signalant que la suite est refusée car non identifiante et demande une réintroduction.
 - # si la suite introduite fait plus de 6 caractères, un message d'erreur est affiché signalant le refus de la suite et demande une réintroduction.
 - # si la suite introduite est vide et que la première est non vide, un message d'erreur est imprimé demandant d'introduire une suite identifiante non vide.
 - # si non, la suite est acceptée.

B) Dans le cas d'un test de type liaison de paragraphes :

.....

Les matériaux sont ici les liaisons flèches. Une liaison flèche au moins doit être pourvue d'une signification.

Une liaison flèche est formée de deux suites de caractères qui l'identifient chacune.

Deux messages vont se succéder tant que vous n'aurez pas donné le signal de fin. (ce signal de fin est l'introduction d'une première suite de caractères vide)

- * le premier message demande l'introduction de la suite identifiante de caractères qui contient la signification en langue naturelle d'une liaison (20 caractères maximum).
- # si la suite introduite est non identifiante par rapport aux liaisons et concepts déjà introduits pour ce test, un message d'erreur est affiché signalant que la suite est refusée car non identifiante et demande une réintroduction.
- # si la suite introduite fait plus de 20 caractères, un message d'erreur est affiché signalant le refus de la suite et demande une réintroduction.
- # si la suite introduite est vide, l'introduction des liaisons est terminée.

si non, la suite est acceptée.

* le second message demande l'introduction de la suite identifiante de caractères qui contient un rappel mnémotechnique de la signification de la liaison (6 caractères maximum);

si la suite introduite est non identifiante par rapport aux concepts et aux liaisons déjà introduits pour ce test, un message d'erreur est affiché signalant que la suite est refusée car non identifiante et demande une réintroduction.

si la suite introduite fait plus de 6 caractères, un message d'erreur est affiché signalant le refus de la suite et demande une réintroduction.

si la suite introduite est vide et que la première est non vide, un message d'erreur est imprimé demandant d'introduire une suite identifiante non vide.

si non, la suite est acceptée.

2.4) Introduction de l'exemple de réponse

Cette partie n'a pas été implémentée. La marche-à-suivre pour introduire un exemple de réponse n'est donc pas expliquée dans cette version du manuel.

2.5) Introduction de la réponse-type

Cette partie n'a pas été implémentée. La marche-à-suivre pour introduire une réponse-type n'est donc pas expliquée dans cette version du manuel.

3) Réalisation d'un test

Cette partie n'a pas été implémentée. La marche-à-suivre pour réaliser un test sera expliquée dans une version ultérieure du manuel.

4) Analyse

Cette partie n'a pas été implémentée. La marche-à-suivre pour réaliser un test sera expliquée dans une version ultérieure du manuel.

DEUXIEME PARTIE : LISTE DES MANIPULATIONS DISPONIBLES

1) Manipulations de consultation d'un texte :

- demande d'accès au texte : cliquez t_txt
- fin de la consultation : cliquez arret
- accès à la page suivante : cliquez p.après
- accès à la page précédente : cliquez p.avant

2) Manipulations de création d'un texte :

- indiquer le soulignement d'une partie de texte :
tapez ^ au début et à la fin de la partie à souligner.
- indiquer l'arrêt de l'introduction du texte :
introduire un corps de paragraphe vide
- indiquer la demande de correction :
(à réaliser !!!)

3) Manipulations de consultation du texte de travail avec insertion possible d'identifiants de paragraphes.

- demande d'accès au texte : cliquez t_txt
- fin de la consultation : cliquez arret
- accès à la page suivante : cliquez p.après
- accès à la page précédente : cliquez p.avant
- introduction d'un identifiant pour un paragraphe donné :
cliquer dans la zone identifiante vide de ce paragraphe (si un identifiant a déjà été donné au paragraphe choisi, un autre identifiant ne peut être introduit) et introduisez l'identifiant.
- choix d'un paragraphe pour l'insérer dans la réponse :
cliquez la zone contenant l'identifiant de paragraphe. Si aucun identifiant n'a été attribué au paragraphe choisi, veuillez d'abord en introduire un avant de choisir le paragraphe.

4) Manipulations sur les concepts lors de la réalisation d'une réponse

- vision des concepts disponibles : cliquez cpt
- choix d'un concept : cliquez le concept choisi
- arrêt de la consultation des concepts : cliquez arrêt
- placement d'un concept dans la réponse : après avoir choisi un concept, cliquez arrêt et positionnez vous à l'endroit ou doit être inséré le concept puis cliquez. Le concept apparaîtra dans la réponse

5) Manipulations sur les liaisons lors de la création d'une réponse

- vision des liaisons disponibles : cliquez liai
- choix d'une liaison : cliquez la liaison choisie
- arrêt de la consultation des liaisons : cliquez arrêt
- placement d'une liaison flèche dans la réponse : après avoir choisi une liaison, cliquez arrêt. Positionnez vous sur le concept origine de la liaison choisie, cliquez puis positionnez vous sur le concept cible et cliquez.
- placement d'une liaison rectangle : à définir !!!!!

6) Manipulations de consultation d'une réponse

- fin de la consultation : cliquez arrêt
- accès à la page suivante : cliquez p.après
- accès à la page précédente : cliquez p.avant

BIBLIOGRAPHIE

Références complètes du document cité en page six :

- " GRILLE D'ANALYSE DE DIDACTICIELS " J. Donnay et M. Romainville
Formation, Recherche en éducation N°1.1. Centre O.S.E
Unité de pédagogie. F.N.D.P Namur (version provisoire Juin 1984)